Variational Inference: Scalability and Recent Advances

Piyush Rai

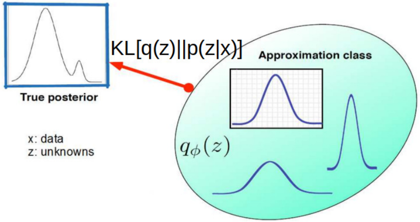
Topics in Probabilistic Modeling and Inference (CS698X)

Feb 25, 2019

Recap: Variational Inference (VI)

- Approximate an intractable posterior $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$ by solving

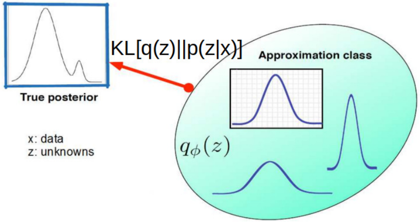
$$\phi^* = \arg\min_{\phi} \text{KL}[q_\phi(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X})] \quad \text{or equivalently} \quad q^*(\mathbf{Z}) = \arg\min_{q \in \mathcal{Q}} \text{KL}[q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X})]$$



Recap: Variational Inference (VI)

- Approximate an intractable posterior $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$ by solving

$$\phi^* = \arg\min_{\phi} \text{KL}[q_\phi(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X})] \quad \text{or equivalently} \quad q^*(\mathbf{Z}) = \arg\min_{q\in\mathcal{Q}} \text{KL}[q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X})]$$

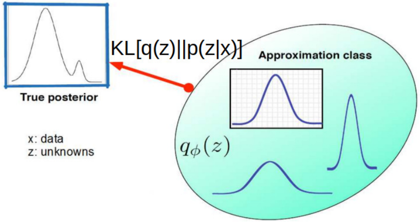


- Equivalent to finding q that maximizes the Evidence Lower Bound (ELBO)

Recap: Variational Inference (VI)

- Approximate an intractable posterior $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$ by solving

$$\phi^* = \arg\min_{\phi} \text{KL}[q_\phi(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X})] \quad \text{or equivalently} \quad q^*(\mathbf{Z}) = \arg\min_{q\in\mathcal{Q}} \text{KL}[q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X})]$$



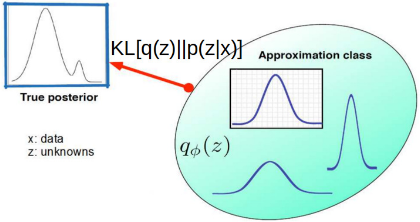
- Equivalent to finding q that maximizes the Evidence Lower Bound (ELBO)

$$\mathcal{L}(q) = \mathcal{L}(\phi) \quad = \quad \mathbb{E}_q[\log p(\mathbf{X},\mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})] = \mathbb{E}_q[\log p(\mathbf{X}|\mathbf{Z})] - \text{KL}(q(\mathbf{Z})||p(\mathbf{Z}))$$

Recap: Variational Inference (VI)

- Approximate an intractable posterior $p(\mathbf{Z}|\mathbf{X})$ by another distribution $q(\mathbf{Z}|\phi)$ by solving

$$\phi^* = \arg\min_{\phi} \text{KL}[q_\phi(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X})] \quad \text{or equivalently} \quad q^*(\mathbf{Z}) = \arg\min_{q\in\mathcal{Q}} \text{KL}[q(\mathbf{Z})||p(\mathbf{Z}|\mathbf{X})]$$



- Equivalent to finding q that maximizes the Evidence Lower Bound (ELBO)

$$\mathcal{L}(q) = \mathcal{L}(\phi) \quad = \quad \mathbb{E}_q[\log p(\mathbf{X},\mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})] = \mathbb{E}_q[\log p(\mathbf{X}|\mathbf{Z})] - \text{KL}(q(\mathbf{Z})||p(\mathbf{Z}))$$

- VI requires solving an **optimization problem** in general (but closed-form solution exists in some special cases, e.g., mean-field VI in locally-conjugate models)

Recap: Mean-Field VI

- Mean-Field VI: Assume $q(\mathbf{Z}|\phi) = \prod_{j=1}^{M} q(\mathbf{Z}_j|\phi_j) = \prod_{j=1}^{N} q_j(\mathbf{Z}_j)$

Recap: Mean-Field VI

- Mean-Field VI: Assume $q(\mathbf{Z}|\phi) = \prod_{j=1}^{M} q(\mathbf{Z}_j|\phi_j) = \prod_{j=1}^{N} q_j(\mathbf{Z}_j)$
- For the optimal q_j, $\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i \neq j}[\ln p(\mathbf{X}, \mathbf{Z})] + \text{const}$

Recap: Mean-Field VI

- Mean-Field VI: Assume $q(\mathbf{Z}|\phi) = \prod_{j=1}^{M} q(\mathbf{Z}_j|\phi_j) = \prod_{j=1}^{N} q_j(\mathbf{Z}_j)$
- For the optimal q_j, $\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})] + \text{const}$, and thus

$$q_j^*(\mathbf{Z}_j) = \frac{\exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})])}{\int \exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})])d\mathbf{Z}_j} \propto \exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})]) \qquad \forall j$$

Recap: Mean-Field VI

- Mean-Field VI: Assume $q(\mathbf{Z}|\phi) = \prod_{j=1}^{M} q(\mathbf{Z}_j|\phi_j) = \prod_{j=1}^{N} q_j(\mathbf{Z}_j)$
- For the optimal q_j, $\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i \neq j}[\ln p(\mathbf{X}, \mathbf{Z})] + \text{const}$, and thus

$$q_j^*(\mathbf{Z}_j) = \frac{\exp(\mathbb{E}_{i \neq j}[\ln p(\mathbf{X}, \mathbf{Z})])}{\int \exp(\mathbb{E}_{i \neq j}[\ln p(\mathbf{X}, \mathbf{Z})]) d\mathbf{Z}_j} \propto \exp(\mathbb{E}_{i \neq j}[\ln p(\mathbf{X}, \mathbf{Z})]) \qquad \forall j$$

- We can also write $\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i \neq j}[\log p(\mathbf{Z}_j|\mathbf{X}, \mathbf{Z}_{-j})] + \text{const}$

Recap: Mean-Field VI

- Mean-Field VI: Assume $q(\mathbf{Z}|\phi) = \prod_{j=1}^{M} q(\mathbf{Z}_j|\phi_j) = \prod_{j=1}^{N} q_j(\mathbf{Z}_j)$
- For the optimal q_j, $\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})] + \text{const}$, and thus

$$q_j^*(\mathbf{Z}_j) = \frac{\exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})])}{\int \exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})])d\mathbf{Z}_j} \propto \exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})]) \qquad \forall j$$

- We can also write $\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i\neq j}[\log p(\mathbf{Z}_j|\mathbf{X},\mathbf{Z}_{-j})] + \text{const}$
- For locally conjugate models, the CP $p(\mathbf{Z}_j|\mathbf{X},\mathbf{Z}_{-j})$ is easy to find, and usually an exp-fam dist.

$$p(\mathbf{Z}_j|\mathbf{X},\mathbf{Z}_{-j}) = h(\mathbf{Z}_j)\exp\left[\eta(\mathbf{X},\mathbf{Z}_{-j})^\top \mathbf{Z}_j - A(\eta(\mathbf{X},\mathbf{Z}_{-j}))\right]$$

Recap: Mean-Field VI

- Mean-Field VI: Assume $q(\mathbf{Z}|\phi) = \prod_{j=1}^{M} q(\mathbf{Z}_j|\phi_j) = \prod_{j=1}^{N} q_j(\mathbf{Z}_j)$
- For the optimal q_j, $\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})] + \text{const}$, and thus

$$q_j^*(\mathbf{Z}_j) = \frac{\exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})])}{\int \exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})])d\mathbf{Z}_j} \propto \exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})]) \qquad \forall j$$

- We can also write $\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i\neq j}[\log p(\mathbf{Z}_j|\mathbf{X},\mathbf{Z}_{-j})] + \text{const}$
- For locally conjugate models, the CP $p(\mathbf{Z}_j|\mathbf{X},\mathbf{Z}_{-j})$ is easy to find, and usually an exp-fam dist.

$$p(\mathbf{Z}_j|\mathbf{X},\mathbf{Z}_{-j}) = h(\mathbf{Z}_j)\exp\left[\eta(\mathbf{X},\mathbf{Z}_{-j})^\top \mathbf{Z}_j - A(\eta(\mathbf{X},\mathbf{Z}_{-j}))\right]$$

- In such a case, each optimal mean-field distribution will be of the form

$$q_j^*(\mathbf{Z}_j) \propto h(\mathbf{Z}_j)\exp\left[\mathbb{E}_{i\neq j}[\eta(\mathbf{X},\mathbf{Z}_{-j})]^\top \mathbf{Z}_j\right]$$

Recap: Mean-Field VI

- Mean-Field VI: Assume $q(\mathbf{Z}|\phi) = \prod_{j=1}^{M} q(\mathbf{Z}_j|\phi_j) = \prod_{j=1}^{N} q_j(\mathbf{Z}_j)$
- For the optimal q_j, $\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})] + \text{const}$, and thus

$$q_j^*(\mathbf{Z}_j) = \frac{\exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})])}{\int \exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})])d\mathbf{Z}_j} \propto \exp(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})]) \qquad \forall j$$

- We can also write $\log q_j^*(\mathbf{Z}_j) = \mathbb{E}_{i\neq j}[\log p(\mathbf{Z}_j|\mathbf{X},\mathbf{Z}_{-j})] + \text{const}$
- For locally conjugate models, the CP $p(\mathbf{Z}_j|\mathbf{X},\mathbf{Z}_{-j})$ is easy to find, and usually an exp-fam dist.

$$p(\mathbf{Z}_j|\mathbf{X},\mathbf{Z}_{-j}) = h(\mathbf{Z}_j)\exp\left[\eta(\mathbf{X},\mathbf{Z}_{-j})^\top \mathbf{Z}_j - A(\eta(\mathbf{X},\mathbf{Z}_{-j}))\right]$$

- In such a case, each optimal mean-field distribution will be of the form

$$q_j^*(\mathbf{Z}_j) \propto h(\mathbf{Z}_j)\exp\left[\mathbb{E}_{i\neq j}[\eta(\mathbf{X},\mathbf{Z}_{-j})]^\top \mathbf{Z}_j\right]$$

.. so its parameters $\phi_j = \mathbb{E}_{i\neq j}[\eta(\mathbf{X},\mathbf{Z}_{-j})]$, i.e., expectation of the natural params of the CP

Recap: VI for Models with Local and Global Variables

- Assuming independence, data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, and local and global unknowns $\mathbf{Z}, \boldsymbol{\beta}$, their joint

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n | \boldsymbol{z}_n, \boldsymbol{\beta}) p(\boldsymbol{z}_n | \boldsymbol{\beta})$$

Recap: VI for Models with Local and Global Variables

- Assuming independence, data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, and local and global unknowns $\mathbf{Z}, \boldsymbol{\beta}$, their joint

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n | \boldsymbol{z}_n, \boldsymbol{\beta}) p(\boldsymbol{z}_n | \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta})$$

Recap: VI for Models with Local and Global Variables

- Assuming independence, data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, and local and global unknowns $\mathbf{Z}, \boldsymbol{\beta}$, their joint

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n | \boldsymbol{z}_n, \boldsymbol{\beta}) p(\boldsymbol{z}_n | \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta})$$

- Assume all the distributions in the above to be **exp-family distributions**

$$p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta}) = h(\boldsymbol{x}_n, \boldsymbol{z}_n) \exp\left[\boldsymbol{\beta}^\top t(\boldsymbol{x}_n, \boldsymbol{z}_n) - A(\boldsymbol{\beta})\right], p(\boldsymbol{\beta} | \boldsymbol{\alpha}) = h(\boldsymbol{\beta}) \exp\left[\boldsymbol{\alpha}^\top [\boldsymbol{\beta}, -A(\boldsymbol{\beta})] - A(\boldsymbol{\alpha})\right]$$

Recap: VI for Models with Local and Global Variables

- Assuming independence, data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, and local and global unknowns $\mathbf{Z}, \boldsymbol{\beta}$, their joint

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n | \boldsymbol{z}_n, \boldsymbol{\beta}) p(\boldsymbol{z}_n | \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta})$$

- Assume all the distributions in the above to be **exp-family distributions**

$$p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta}) = h(\boldsymbol{x}_n, \boldsymbol{z}_n) \exp\left[\boldsymbol{\beta}^\top t(\boldsymbol{x}_n, \boldsymbol{z}_n) - A(\boldsymbol{\beta})\right], p(\boldsymbol{\beta}|\boldsymbol{\alpha}) = h(\boldsymbol{\beta}) \exp\left[\boldsymbol{\alpha}^\top [\boldsymbol{\beta}, -A(\boldsymbol{\beta})] - A(\boldsymbol{\alpha})\right]$$

- Also assuming $p(\boldsymbol{x}_n | \boldsymbol{z}_n)$ and $p(\boldsymbol{z}_n)$ to be conjugate, CPs for $\boldsymbol{z}_n$ and $\boldsymbol{\beta}$ are also exp-fam

$$p(\boldsymbol{z}_n | \boldsymbol{x}_n, \boldsymbol{\beta}) \quad \propto \quad h(\boldsymbol{z}_n) \exp\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})^\top \boldsymbol{z}_n\right]$$

Recap: VI for Models with Local and Global Variables

- Assuming independence, data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, and local and global unknowns $\mathbf{Z}, \boldsymbol{\beta}$, their joint

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n | \boldsymbol{z}_n, \boldsymbol{\beta}) p(\boldsymbol{z}_n | \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta})$$

- Assume all the distributions in the above to be **exp-family distributions**

$$p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta}) = h(\boldsymbol{x}_n, \boldsymbol{z}_n) \exp\left[\boldsymbol{\beta}^\top t(\boldsymbol{x}_n, \boldsymbol{z}_n) - A(\boldsymbol{\beta})\right], p(\boldsymbol{\beta} | \boldsymbol{\alpha}) = h(\boldsymbol{\beta}) \exp\left[\boldsymbol{\alpha}^\top [\boldsymbol{\beta}, -A(\boldsymbol{\beta})] - A(\boldsymbol{\alpha})\right]$$

- Also assuming $p(\boldsymbol{x}_n | \boldsymbol{z}_n)$ and $p(\boldsymbol{z}_n)$ to be conjugate, CPs for $\boldsymbol{z}_n$ and $\boldsymbol{\beta}$ are also exp-fam

$$\begin{aligned} p(\boldsymbol{z}_n | \boldsymbol{x}_n, \boldsymbol{\beta}) &\propto h(\boldsymbol{z}_n) \exp\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})^\top \boldsymbol{z}_n\right] \\ p(\boldsymbol{\beta} | \mathbf{X}, \mathbf{Z}) &\propto h(\boldsymbol{\beta}) \exp\left[[\alpha_1 + \sum_{n=1}^{N} t(\boldsymbol{x}_n, \boldsymbol{z}_n), \alpha_2 + N]^\top [\boldsymbol{\beta}, -A(\boldsymbol{\beta})]\right] \end{aligned}$$

Recap: VI for Models with Local and Global Variables

- Assuming independence, data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, and local and global unknowns $\mathbf{Z}, \boldsymbol{\beta}$, their joint

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n | \boldsymbol{z}_n, \boldsymbol{\beta}) p(\boldsymbol{z}_n | \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta})$$

- Assume all the distributions in the above to be **exp-family distributions**

$$p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta}) = h(\boldsymbol{x}_n, \boldsymbol{z}_n) \exp\left[\boldsymbol{\beta}^\top t(\boldsymbol{x}_n, \boldsymbol{z}_n) - A(\boldsymbol{\beta})\right], p(\boldsymbol{\beta}|\boldsymbol{\alpha}) = h(\boldsymbol{\beta}) \exp\left[\boldsymbol{\alpha}^\top [\boldsymbol{\beta}, -A(\boldsymbol{\beta})] - A(\boldsymbol{\alpha})\right]$$

- Also assuming $p(\boldsymbol{x}_n|\boldsymbol{z}_n)$ and $p(\boldsymbol{z}_n)$ to be conjugate, CPs for $\boldsymbol{z}_n$ and $\boldsymbol{\beta}$ are also exp-fam

$$p(\boldsymbol{z}_n | \boldsymbol{x}_n, \boldsymbol{\beta}) \quad \propto \quad h(\boldsymbol{z}_n) \exp\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})^\top \boldsymbol{z}_n\right]$$

$$p(\boldsymbol{\beta} | \mathbf{X}, \mathbf{Z}) \quad \propto \quad h(\boldsymbol{\beta}) \exp\left[[\alpha_1 + \sum_{n=1}^{N} t(\boldsymbol{x}_n, \boldsymbol{z}_n), \alpha_2 + N]^\top [\boldsymbol{\beta}, -A(\boldsymbol{\beta})]\right]$$

- Assuming $q(\boldsymbol{\beta}, \mathbf{Z}) = q(\boldsymbol{\beta}|\lambda) \prod_{n=1}^{N} q(\boldsymbol{z}_n|\phi_n)$, the optimal local and global var. params

$$\phi_n = \mathbb{E}_\lambda [\eta(\boldsymbol{x}_n, \boldsymbol{\beta})] \qquad \forall n$$

Recap: VI for Models with Local and Global Variables

- Assuming independence, data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, and local and global unknowns $\mathbf{Z}, \boldsymbol{\beta}$, their joint

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n | \boldsymbol{z}_n, \boldsymbol{\beta}) p(\boldsymbol{z}_n | \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta})$$

- Assume all the distributions in the above to be **exp-family distributions**

$$p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta}) = h(\boldsymbol{x}_n, \boldsymbol{z}_n) \exp\left[\boldsymbol{\beta}^\top t(\boldsymbol{x}_n, \boldsymbol{z}_n) - A(\boldsymbol{\beta})\right], p(\boldsymbol{\beta}|\boldsymbol{\alpha}) = h(\boldsymbol{\beta}) \exp\left[\boldsymbol{\alpha}^\top [\boldsymbol{\beta}, -A(\boldsymbol{\beta})] - A(\boldsymbol{\alpha})\right]$$

- Also assuming $p(\boldsymbol{x}_n | \boldsymbol{z}_n)$ and $p(\boldsymbol{z}_n)$ to be conjugate, CPs for $\boldsymbol{z}_n$ and $\boldsymbol{\beta}$ are also exp-fam

$$p(\boldsymbol{z}_n | \boldsymbol{x}_n, \boldsymbol{\beta}) \propto h(\boldsymbol{z}_n) \exp\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})^\top \boldsymbol{z}_n\right]$$

$$p(\boldsymbol{\beta} | \mathbf{X}, \mathbf{Z}) \propto h(\boldsymbol{\beta}) \exp\left[[\alpha_1 + \sum_{n=1}^{N} t(\boldsymbol{x}_n, \boldsymbol{z}_n), \alpha_2 + N]^\top [\boldsymbol{\beta}, -A(\boldsymbol{\beta})]\right]$$

- Assuming $q(\boldsymbol{\beta}, \mathbf{Z}) = q(\boldsymbol{\beta}|\lambda) \prod_{n=1}^{N} q(\boldsymbol{z}_n | \phi_n)$, the optimal local and global var. params

$$\phi_n = \mathbb{E}_\lambda [\eta(\boldsymbol{x}_n, \boldsymbol{\beta})] \quad \forall n, \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N} \mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\boldsymbol{\alpha}}]$$

Recap: VI for Models with Local and Global Variables

- Assuming independence, data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, and local and global unknowns $\mathbf{Z}, \boldsymbol{\beta}$, their joint

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n | \boldsymbol{z}_n, \boldsymbol{\beta}) p(\boldsymbol{z}_n | \boldsymbol{\beta}) = p(\boldsymbol{\beta}) \prod_{n=1}^{N} p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta})$$

- Assume all the distributions in the above to be **exp-family distributions**

$$p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\beta}) = h(\boldsymbol{x}_n, \boldsymbol{z}_n) \exp\left[\boldsymbol{\beta}^\top t(\boldsymbol{x}_n, \boldsymbol{z}_n) - A(\boldsymbol{\beta})\right], p(\boldsymbol{\beta}|\boldsymbol{\alpha}) = h(\boldsymbol{\beta}) \exp\left[\boldsymbol{\alpha}^\top [\boldsymbol{\beta}, -A(\boldsymbol{\beta})] - A(\boldsymbol{\alpha})\right]$$

- Also assuming $p(\boldsymbol{x}_n | \boldsymbol{z}_n)$ and $p(\boldsymbol{z}_n)$ to be conjugate, CPs for $\boldsymbol{z}_n$ and $\boldsymbol{\beta}$ are also exp-fam

$$p(\boldsymbol{z}_n | \boldsymbol{x}_n, \boldsymbol{\beta}) \propto h(\boldsymbol{z}_n) \exp\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})^\top \boldsymbol{z}_n\right]$$

$$p(\boldsymbol{\beta} | \mathbf{X}, \mathbf{Z}) \propto h(\boldsymbol{\beta}) \exp\left[[\alpha_1 + \sum_{n=1}^{N} t(\boldsymbol{x}_n, \boldsymbol{z}_n), \alpha_2 + N]^\top [\boldsymbol{\beta}, -A(\boldsymbol{\beta})]\right]$$

- Assuming $q(\boldsymbol{\beta}, \mathbf{Z}) = q(\boldsymbol{\beta}|\lambda) \prod_{n=1}^{N} q(\boldsymbol{z}_n | \phi_n)$, the optimal local and global var. params

$$\phi_n = \mathbb{E}_\lambda [\eta(\boldsymbol{x}_n, \boldsymbol{\beta})] \quad \forall n, \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N} \mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\boldsymbol{\alpha}}]$$

- Note: Each update of global var. params requires waiting for all updates of local var. params

Advances in Variational Inference

Plan

- SVI - Stochastic Variational Inference (we'll mainly focus on SVI for locally-conjugate models)

Plan

- SVI - Stochastic Variational Inference (we'll mainly focus on SVI for locally-conjugate models)
- VI/SVI for non-conjugate models
  - Model-specific tricks to handle non-conjugacy
  - Black-Box Variational Inference (BBVI)
  - Reparametrization Trick based VI
  - Automatic Differentiation VI (ADVI) via Unconstrained Optimization

Plan

- SVI - Stochastic Variational Inference (we'll mainly focus on SVI for locally-conjugate models)
- VI/SVI for non-conjugate models
  - Model-specific tricks to handle non-conjugacy
  - Black-Box Variational Inference (BBVI)
  - Reparametrization Trick based VI
  - Automatic Differentiation VI (ADVI) via Unconstrained Optimization
- Amortized Variational Inference

Plan

- SVI - Stochastic Variational Inference (we'll mainly focus on SVI for locally-conjugate models)
- VI/SVI for non-conjugate models
  - Model-specific tricks to handle non-conjugacy
  - Black-Box Variational Inference (BBVI)
  - Reparametrization Trick based VI
  - Automatic Differentiation VI (ADVI) via Unconstrained Optimization
- Amortized Variational Inference
- Structured Variational Inference

Plan

- SVI - Stochastic Variational Inference (we'll mainly focus on SVI for locally-conjugate models)
- VI/SVI for non-conjugate models
  - Model-specific tricks to handle non-conjugacy
  - Black-Box Variational Inference (BBVI)
  - Reparametrization Trick based VI
  - Automatic Differentiation VI (ADVI) via Unconstrained Optimization
- Amortized Variational Inference
- Structured Variational Inference
- Other divergences (recall that VI finds optimal q by minimizing the KL divergence $KL(q||p)$)

Stochastic Variational Inference

Stochastic Variational Inference (SVI)

- An "online" algorithm[†] to speed-up VI for LVMs with local and global variables

[†] Stochastic Variational Inference (Hoffman et al, 2013)

Stochastic Variational Inference (SVI)

- An "online" algorithm[†] to speed-up VI for LVMs with local and global variables
- We saw the mean-field VI updates ($q(\beta, \mathbf{Z}) = q(\beta|\lambda) \prod_{n=1}^{N} q(\boldsymbol{z}_n|\phi_n)$) for such models

$$\phi_n = \mathbb{E}_\lambda \left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})\right] \quad \forall n \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N} \mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\boldsymbol{\alpha}}(\mathbf{X}, \mathbf{Z})]$$

[†] Stochastic Variational Inference (Hoffman et al, 2013)

Stochastic Variational Inference (SVI)

- An "online" algorithm[†] to speed-up VI for LVMs with local and global variables
- We saw the mean-field VI updates ($q(\beta, \mathbf{Z}) = q(\beta|\lambda)\prod_{n=1}^{N} q(\boldsymbol{z}_n|\phi_n)$) for such models

$$\phi_n = \mathbb{E}_\lambda[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})] \quad \forall n \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N} \mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X}, \mathbf{Z})]$$

- SVI makes the global params λ updates more efficient (note that λ depends on all ϕ_n's)

[†] Stochastic Variational Inference (Hoffman et al, 2013)

Stochastic Variational Inference (SVI)

- An "online" algorithm[†] to speed-up VI for LVMs with local and global variables
- We saw the mean-field VI updates $(q(\beta, \mathbf{Z}) = q(\beta|\lambda)\prod_{n=1}^{N} q(\mathbf{z}_n|\phi_n))$ for such models

$$\phi_n = \mathbb{E}_\lambda\left[\eta(\mathbf{x}_n, \boldsymbol{\beta})\right] \quad \forall n \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N} \mathbb{E}_{\phi_n}[t(\mathbf{x}_n, \mathbf{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X}, \mathbf{Z})]$$

- SVI makes the global params λ updates more efficient (note that λ depends on all ϕ_n's)
- SVI works with minibatches of data as follows (assuming minibatch size $= 1$)

[†] Stochastic Variational Inference (Hoffman et al, 2013)

Stochastic Variational Inference (SVI)

- An "online" algorithm[†] to speed-up VI for LVMs with local and global variables
- We saw the mean-field VI updates ($q(\beta, \mathbf{Z}) = q(\beta|\lambda)\prod_{n=1}^{N} q(\mathbf{z}_n|\phi_n)$) for such models

$$\phi_n = \mathbb{E}_\lambda[\eta(\mathbf{x}_n, \boldsymbol{\beta})] \quad \forall n \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N} \mathbb{E}_{\phi_n}[t(\mathbf{x}_n, \mathbf{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\boldsymbol{\alpha}}(\mathbf{X}, \mathbf{Z})]$$

- SVI makes the global params λ updates more efficient (note that λ depends on all ϕ_n's)
- SVI works with minibatches of data as follows (assuming minibatch size $= 1$)
  1. Initialize λ randomly as $\lambda^{(0)}$ and set current iteration number as $i = 1$

[†] Stochastic Variational Inference (Hoffman et al, 2013)

Stochastic Variational Inference (SVI)

- An "online" algorithm[†] to speed-up VI for LVMs with local and global variables
- We saw the mean-field VI updates ($q(\beta, \mathbf{Z}) = q(\beta|\lambda) \prod_{n=1}^{N} q(\boldsymbol{z}_n|\phi_n)$) for such models

$$\phi_n = \mathbb{E}_\lambda [\eta(\boldsymbol{x}_n, \boldsymbol{\beta})] \quad \forall n \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N} \mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X}, \mathbf{Z})]$$

- SVI makes the global params λ updates more efficient (note that λ depends on all ϕ_n's)
- SVI works with minibatches of data as follows (assuming minibatch size = 1)

1. Initialize λ randomly as $\lambda^{(0)}$ and set current iteration number as $i = 1$
2. Set the learning rate (decaying as) as $\epsilon_i = (i+1)^{-\kappa}$ where $\kappa \in (0.5, 1]$

[†] Stochastic Variational Inference (Hoffman et al, 2013)

Stochastic Variational Inference (SVI)

- An "online" algorithm† to speed-up VI for LVMs with local and global variables
- We saw the mean-field VI updates $(q(\beta, \mathbf{Z}) = q(\beta|\lambda)\prod_{n=1}^{N} q(\boldsymbol{z}_n|\phi_n))$ for such models

$$\phi_n = \mathbb{E}_\lambda\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})\right] \quad \forall n \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N}\mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\boldsymbol{\alpha}}(\mathbf{X}, \mathbf{Z})]$$

- SVI makes the global params λ updates more efficient (note that λ depends on all ϕ_n's)
- SVI works with minibatches of data as follows (assuming minibatch size $= 1$)

1. Initialize λ randomly as $\lambda^{(0)}$ and set current iteration number as $i = 1$
2. Set the learning rate (decaying as) as $\epsilon_i = (i+1)^{-\kappa}$ where $\kappa \in (0.5, 1]$
3. Choose a data point n randomly, i.e., $n \sim \text{Uniform}(1, \ldots, N)$

† Stochastic Variational Inference (Hoffman et al, 2013)

Stochastic Variational Inference (SVI)

- An "online" algorithm[†] to speed-up VI for LVMs with local and global variables
- We saw the mean-field VI updates $(q(\beta, \mathbf{Z}) = q(\beta|\lambda)\prod_{n=1}^{N} q(\mathbf{z}_n|\phi_n))$ for such models

$$\phi_n = \mathbb{E}_{\lambda}[\eta(\mathbf{x}_n, \beta)] \quad \forall n \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N}\mathbb{E}_{\phi_n}[t(\mathbf{x}_n, \mathbf{z}_n)], \alpha_2 + N\right]^{\top} = \mathbb{E}_{\phi}[\hat{\alpha}(\mathbf{X}, \mathbf{Z})]$$

- SVI makes the global params λ updates more efficient (note that λ depends on all ϕ_n's)
- SVI works with minibatches of data as follows (assuming minibatch size $= 1$)

1. Initialize λ randomly as $\lambda^{(0)}$ and set current iteration number as $i = 1$
2. Set the learning rate (decaying as) as $\epsilon_i = (i+1)^{-\kappa}$ where $\kappa \in (0.5, 1]$
3. Choose a data point n randomly, i.e., $n \sim \text{Uniform}(1, \ldots, N)$
4. Compute local var. param ϕ_n for data point $\mathbf{x}_n$ as $\phi_n = \mathbb{E}_{\lambda^{(i-1)}}[\eta(\mathbf{x}_n, \beta)]$

[†] Stochastic Variational Inference (Hoffman et al, 2013)

Stochastic Variational Inference (SVI)

- An "online" algorithm[†] to speed-up VI for LVMs with local and global variables
- We saw the mean-field VI updates ($q(\beta, \mathbf{Z}) = q(\beta|\lambda) \prod_{n=1}^{N} q(\boldsymbol{z}_n|\phi_n)$) for such models

$$\phi_n = \mathbb{E}_\lambda\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})\right] \quad \forall n \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N} \mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X}, \mathbf{Z})]$$

- SVI makes the global params λ updates more efficient (note that λ depends on all ϕ_n's)
- SVI works with minibatches of data as follows (assuming minibatch size $= 1$)
  1. Initialize λ randomly as $\lambda^{(0)}$ and set current iteration number as $i = 1$
  2. Set the learning rate (decaying as) as $\epsilon_i = (i+1)^{-\kappa}$ where $\kappa \in (0.5, 1]$
  3. Choose a data point n randomly, i.e., $n \sim \text{Uniform}(1, \ldots, N)$
  4. Compute local var. param ϕ_n for data point $\boldsymbol{x}_n$ as $\phi_n = \mathbb{E}_{\lambda^{(i-1)}}\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})\right]$
  5. Update λ as $\lambda^{(i)} = (1 - \epsilon_i)\lambda^{(i-1)} + \epsilon_i \lambda_n$

[†] Stochastic Variational Inference (Hoffman et al, 2013)

Stochastic Variational Inference (SVI)

- An "online" algorithm[†] to speed-up VI for LVMs with local and global variables
- We saw the mean-field VI updates $(q(\boldsymbol{\beta}, \mathbf{Z}) = q(\boldsymbol{\beta}|\lambda)\prod_{n=1}^{N} q(\boldsymbol{z}_n|\phi_n))$ for such models

$$\phi_n = \mathbb{E}_\lambda\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})\right] \quad \forall n \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N} \mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\boldsymbol{\alpha}}(\mathbf{X}, \mathbf{Z})]$$

- SVI makes the global params λ updates more efficient (note that λ depends on all ϕ_n's)
- SVI works with minibatches of data as follows (assuming minibatch size = 1)
  1. Initialize λ randomly as $\lambda^{(0)}$ and set current iteration number as $i = 1$
  2. Set the learning rate (decaying as) as $\epsilon_i = (i+1)^{-\kappa}$ where $\kappa \in (0.5, 1]$
  3. Choose a data point n randomly, i.e., $n \sim \text{Uniform}(1, \ldots, N)$
  4. Compute local var. param ϕ_n for data point $\boldsymbol{x}_n$ as $\phi_n = \mathbb{E}_{\lambda^{(i-1)}}\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})\right]$
  5. Update λ as $\lambda^{(i)} = (1-\epsilon_i)\lambda^{(i-1)} + \epsilon_i\lambda_n$ where $\lambda_n = [\alpha_1 + \mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + 1]^\top$

[†] Stochastic Variational Inference (Hoffman et al, 2013)

Stochastic Variational Inference (SVI)

- An "online" algorithm[†] to speed-up VI for LVMs with local and global variables
- We saw the mean-field VI updates ($q(\boldsymbol{\beta}, \mathbf{Z}) = q(\boldsymbol{\beta}|\lambda)\prod_{n=1}^{N} q(\boldsymbol{z}_n|\phi_n)$) for such models

$$\phi_n = \mathbb{E}_\lambda\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})\right] \quad \forall n \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N}\mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\boldsymbol{\alpha}}(\mathbf{X}, \mathbf{Z})]$$

- SVI makes the global params λ updates more efficient (note that λ depends on all ϕ_n's)
- SVI works with minibatches of data as follows (assuming minibatch size $= 1$)

1. Initialize λ randomly as $\lambda^{(0)}$ and set current iteration number as $i = 1$
2. Set the learning rate (decaying as) as $\epsilon_i = (i+1)^{-\kappa}$ where $\kappa \in (0.5, 1]$
3. Choose a data point n randomly, i.e., $n \sim \text{Uniform}(1, \ldots, N)$
4. Compute local var. param ϕ_n for data point $\boldsymbol{x}_n$ as $\phi_n = \mathbb{E}_{\lambda^{(i-1)}}\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})\right]$
5. Update λ as $\lambda^{(i)} = (1-\epsilon_i)\lambda^{(i-1)} + \epsilon_i\lambda_n$ where $\lambda_n = [\alpha_1 + \mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + 1]^\top = \mathbb{E}_{\phi_n}[\hat{\boldsymbol{\alpha}}(\boldsymbol{x}_n, \boldsymbol{z}_n)]$

[†] Stochastic Variational Inference (Hoffman et al, 2013)

Stochastic Variational Inference (SVI)

- An "online" algorithm[†] to speed-up VI for LVMs with local and global variables
- We saw the mean-field VI updates ($q(\beta, \mathbf{Z}) = q(\beta|\lambda) \prod_{n=1}^{N} q(\boldsymbol{z}_n|\phi_n)$) for such models

$$\phi_n = \mathbb{E}_\lambda\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})\right] \quad \forall n \quad \text{and} \quad \lambda = \left[\alpha_1 + \sum_{n=1}^{N} \mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + N\right]^\top = \mathbb{E}_\phi[\hat{\boldsymbol{\alpha}}(\mathbf{X}, \mathbf{Z})]$$

- SVI makes the global params λ updates more efficient (note that λ depends on all ϕ_n's)
- SVI works with minibatches of data as follows (assuming minibatch size $= 1$)

1. Initialize λ randomly as $\lambda^{(0)}$ and set current iteration number as $i = 1$
2. Set the learning rate (decaying as) as $\epsilon_i = (i+1)^{-\kappa}$ where $\kappa \in (0.5, 1]$
3. Choose a data point n randomly, i.e., $n \sim \text{Uniform}(1, \ldots, N)$
4. Compute local var. param ϕ_n for data point $\boldsymbol{x}_n$ as $\phi_n = \mathbb{E}_{\lambda^{(i-1)}}\left[\eta(\boldsymbol{x}_n, \boldsymbol{\beta})\right]$
5. Update λ as $\lambda^{(i)} = (1 - \epsilon_i)\lambda^{(i-1)} + \epsilon_i \lambda_n$ where $\lambda_n = [\alpha_1 + \mathbb{E}_{\phi_n}[t(\boldsymbol{x}_n, \boldsymbol{z}_n)], \alpha_2 + 1]^\top = \mathbb{E}_{\phi_n}[\hat{\boldsymbol{\alpha}}(\boldsymbol{x}_n, \boldsymbol{z}_n)]$
6. Set $i = i + 1$. If ELBO not converged, go to Step 2

[†] Stochastic Variational Inference (Hoffman et al, 2013)

What is SVI Doing?

- SVI updates the global var params λ using stochastic optimization of the ELBO[†]

[†] Stochastic Variational Inference (Hoffman et al, 2013)

What is SVI Doing?

- SVI updates the global var params λ using stochastic optimization of the ELBO[†]
- Instead of usual gradient of ELBO w.r.t. λ, SVI uses the natural gradient

[†] Stochastic Variational Inference (Hoffman et al, 2013)

What is SVI Doing?

- SVI updates the global var params λ using stochastic optimization of the ELBO[†]
- Instead of usual gradient of ELBO w.r.t. λ, SVI uses the natural gradient
  - Denoting the double derivative of the log-partition function of CP of β as A''

[†] Stochastic Variational Inference (Hoffman et al, 2013)

What is SVI Doing?

- SVI updates the global var params λ using stochastic optimization of the ELBO†
- Instead of usual gradient of ELBO w.r.t. λ, SVI uses the natural gradient
  - Denoting the double derivative of the log-partition function of CP of β as A''

$$\text{Usual gradient:} \quad \nabla_\lambda \text{ELBO} = A''(\lambda)(\mathbb{E}_\phi[\hat{\alpha}(\mathbf{X},\mathbf{Z})] - \lambda) \qquad \text{(exercise)}$$

† Stochastic Variational Inference (Hoffman et al, 2013)

What is SVI Doing?

- SVI updates the global var params λ using stochastic optimization of the ELBO[†]
- Instead of usual gradient of ELBO w.r.t. λ, SVI uses the natural gradient
  - Denoting the double derivative of the log-partition function of CP of β as A''

$$\begin{aligned}\text{Usual gradient:} \quad \nabla_\lambda \text{ELBO} &= A''(\lambda)(\mathbb{E}_\phi[\hat{\alpha}(\mathbf{X},\mathbf{Z})] - \lambda) \qquad \text{(exercise)} \\ \text{Natural gradient:} \quad g(\lambda) &= A''(\lambda)^{-1} \times \nabla_\lambda \text{ELBO} = \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X},\mathbf{Z})] - \lambda\end{aligned}$$

† Stochastic Variational Inference (Hoffman et al, 2013)

What is SVI Doing?

- SVI updates the global var params λ using stochastic optimization of the ELBO[†]
- Instead of usual gradient of ELBO w.r.t. λ, SVI uses the natural gradient
  - Denoting the double derivative of the log-partition function of CP of β as A''

$$\text{Usual gradient:} \quad \nabla_\lambda \text{ELBO} = A''(\lambda)(\mathbb{E}_\phi[\hat{\alpha}(\mathbf{X},\mathbf{Z})] - \lambda) \qquad \text{(exercise)}$$

$$\text{Natural gradient:} \quad g(\lambda) = A''(\lambda)^{-1} \times \nabla_\lambda \text{ELBO} = \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X},\mathbf{Z})] - \lambda$$

  - Note: $A''(\lambda)$ is cov. of suff-stats of CP of β and $A''(\lambda)^{-1}$ is the Fisher information matrix

[†] Stochastic Variational Inference (Hoffman et al, 2013)

What is SVI Doing?

- SVI updates the global var params λ using stochastic optimization of the ELBO†
- Instead of usual gradient of ELBO w.r.t. λ, SVI uses the natural gradient
  - Denoting the double derivative of the log-partition function of CP of β as A''

$$
\begin{aligned}
\text{Usual gradient:} \quad \nabla_\lambda \text{ELBO} &= A''(\lambda)(\mathbb{E}_\phi[\hat{\alpha}(\mathbf{X}, \mathbf{Z})] - \lambda) \qquad \text{(exercise)} \\
\text{Natural gradient:} \quad g(\lambda) &= A''(\lambda)^{-1} \times \nabla_\lambda \text{ELBO} = \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X}, \mathbf{Z})] - \lambda
\end{aligned}
$$

  - Note: $A''(\lambda)$ is cov. of suff-stats of CP of β and $A''(\lambda)^{-1}$ is the Fisher information matrix
- Using the natural gradient has some nice advantages

† Stochastic Variational Inference (Hoffman et al, 2013)

What is SVI Doing?

- SVI updates the global var params λ using stochastic optimization of the ELBO[†]
- Instead of usual gradient of ELBO w.r.t. λ, SVI uses the natural gradient
  - Denoting the double derivative of the log-partition function of CP of β as A''

$$\begin{aligned}\text{Usual gradient:} \quad \nabla_\lambda \text{ELBO} &= A''(\lambda)(\mathbb{E}_\phi[\hat{\alpha}(\mathbf{X},\mathbf{Z})] - \lambda) \qquad \text{(exercise)} \\ \text{Natural gradient:} \quad g(\lambda) &= A''(\lambda)^{-1} \times \nabla_\lambda \text{ELBO} = \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X},\mathbf{Z})] - \lambda\end{aligned}$$

  - Note: $A''(\lambda)$ is cov. of suff-stats of CP of β and $A''(\lambda)^{-1}$ is the Fisher information matrix
- Using the natural gradient has some nice advantages
  - Nat. gradient based updates of λ have simple form + easy to compute (no need to compute $A''(\lambda)$)

[†] Stochastic Variational Inference (Hoffman et al, 2013)

What is SVI Doing?

- SVI updates the global var params λ using stochastic optimization of the ELBO[†]
- Instead of usual gradient of ELBO w.r.t. λ, SVI uses the natural gradient
  - Denoting the double derivative of the log-partition function of CP of β as A''

$$\begin{aligned} \text{Usual gradient:} \quad \nabla_\lambda \text{ELBO} &= A''(\lambda)(\mathbb{E}_\phi[\hat{\alpha}(\mathbf{X}, \mathbf{Z})] - \lambda) \qquad \text{(exercise)} \\ \text{Natural gradient:} \quad g(\lambda) &= A''(\lambda)^{-1} \times \nabla_\lambda \text{ELBO} = \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X}, \mathbf{Z})] - \lambda \end{aligned}$$

  - Note: $A''(\lambda)$ is cov. of suff-stats of CP of β and $A''(\lambda)^{-1}$ is the Fisher information matrix
- Using the natural gradient has some nice advantages
  - Nat. gradient based updates of λ have simple form + easy to compute (no need to compute $A''(\lambda)$)

$$\lambda^{(i)} = \lambda^{(i-1)} + \epsilon_i g(\lambda)|_{\lambda^{(i-1)}}$$

[†] Stochastic Variational Inference (Hoffman et al, 2013)

What is SVI Doing?

- SVI updates the global var params λ using stochastic optimization of the ELBO[†]
- Instead of usual gradient of ELBO w.r.t. λ, SVI uses the natural gradient
  - Denoting the double derivative of the log-partition function of CP of β as A''

$$\begin{aligned}\text{Usual gradient:} \quad \nabla_\lambda \text{ELBO} &= A''(\lambda)(\mathbb{E}_\phi[\hat{\alpha}(\mathbf{X}, \mathbf{Z})] - \lambda) \qquad \text{(exercise)}\\ \text{Natural gradient:} \quad g(\lambda) &= A''(\lambda)^{-1} \times \nabla_\lambda \text{ELBO} = \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X}, \mathbf{Z})] - \lambda\end{aligned}$$

  - Note: $A''(\lambda)$ is cov. of suff-stats of CP of β and $A''(\lambda)^{-1}$ is the Fisher information matrix
- Using the natural gradient has some nice advantages
  - Nat. gradient based updates of λ have simple form + easy to compute (no need to compute $A''(\lambda)$)

$$\lambda^{(i)} = \lambda^{(i-1)} + \epsilon_i g(\lambda)|_{\lambda^{(i-1)}} = (1 - \epsilon_i)\lambda^{(i-1)} + \epsilon_i \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X}, \mathbf{Z})] \quad \text{(assuming full batch)}$$

[†] Stochastic Variational Inference (Hoffman et al, 2013)

What is SVI Doing?

- SVI updates the global var params λ using stochastic optimization of the ELBO[†]
- Instead of usual gradient of ELBO w.r.t. λ, SVI uses the natural gradient
  - Denoting the double derivative of the log-partition function of CP of β as A''

$$\text{Usual gradient:} \quad \nabla_\lambda \text{ELBO} = A''(\lambda)(\mathbb{E}_\phi[\hat{\alpha}(\mathbf{X},\mathbf{Z})] - \lambda) \qquad \text{(exercise)}$$

$$\text{Natural gradient:} \quad g(\lambda) = A''(\lambda)^{-1} \times \nabla_\lambda \text{ELBO} = \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X},\mathbf{Z})] - \lambda$$

  - Note: $A''(\lambda)$ is cov. of suff-stats of CP of β and $A''(\lambda)^{-1}$ is the Fisher information matrix
- Using the natural gradient has some nice advantages
  - Nat. gradient based updates of λ have simple form + easy to compute (no need to compute $A''(\lambda)$)

$$\lambda^{(i)} = \lambda^{(i-1)} + \epsilon_i g(\lambda)|_{\lambda^{(i-1)}} = (1-\epsilon_i)\lambda^{(i-1)} + \epsilon_i \mathbb{E}_\phi[\hat{\alpha}(\mathbf{X},\mathbf{Z})] \quad \text{(assuming full batch)}$$

  - Natural gradients are more intuitive/meaningful: Euclidean distance isn't often meaningful when used to compute distance between parameters of probability distributions, e.g., $q(\beta|\lambda)$ and $q(\beta|\lambda')$

[†] Stochastic Variational Inference (Hoffman et al, 2013)

SVI: Some Comments

- Often operates on minibatches: For iteration i minibatch $\mathcal{B}_i$, update λ as follows

$$\hat{\lambda} = \frac{1}{|\mathcal{B}_i|}\sum_{n\in\mathcal{B}_i}\lambda_n$$

$$\lambda^{(i)} = (1-\epsilon_i)\lambda^{(i-1)} + \epsilon_i\hat{\lambda}$$

† Stochastic Variational Inference (Hoffman et al, 2013)

SVI: Some Comments

- Often operates on minibatches: For iteration i minibatch $\mathcal{B}_i$, update λ as follows

$$\hat{\lambda} = \frac{1}{|\mathcal{B}_i|}\sum_{n\in\mathcal{B}_i}\lambda_n$$

$$\lambda^{(i)} = (1-\epsilon_i)\lambda^{(i-1)} + \epsilon_i\hat{\lambda}$$

- Decaying learning rate is necessary for convergence (need $\sum_i \epsilon_i = \infty$ and $\sum_i \epsilon_i^2 < \infty$)

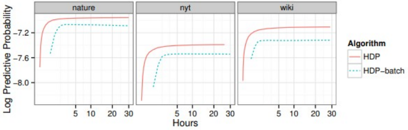
† Stochastic Variational Inference (Hoffman et al, 2013)

SVI: Some Comments

- Often operates on minibatches: For iteration i minibatch $\mathcal{B}_i$, update λ as follows

$$\hat{\lambda} = \frac{1}{|\mathcal{B}_i|}\sum_{n\in\mathcal{B}_i}\lambda_n$$
$$\lambda^{(i)} = (1-\epsilon_i)\lambda^{(i-1)} + \epsilon_i\hat{\lambda}$$

- Decaying learning rate is necessary for convergence (need $\sum_i \epsilon_i = \infty$ and $\sum_i \epsilon_i^2 < \infty$)
- SVI successfully used on many large-scale problems (document topic modeling, citation network analysis, etc). Often has much faster convergence (and better results) as compared to batch VI



SVI vs Batch VI on a nonparametric Bayesian Topic Model (Hierarchical Dirichlet Process)

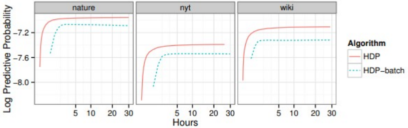
† Stochastic Variational Inference (Hoffman et al, 2013)

SVI: Some Comments

- Often operates on minibatches: For iteration i minibatch $\mathcal{B}_i$, update λ as follows

$$\begin{aligned}\hat{\lambda} &= \frac{1}{|\mathcal{B}_i|}\sum_{n\in\mathcal{B}_i}\lambda_n \\ \lambda^{(i)} &= (1-\epsilon_i)\lambda^{(i-1)} + \epsilon_i\hat{\lambda}\end{aligned}$$

- Decaying learning rate is necessary for convergence (need $\sum_i \epsilon_i = \infty$ and $\sum_i \epsilon_i^2 < \infty$)
- SVI successfully used on many large-scale problems (document topic modeling, citation network analysis, etc). Often has much faster convergence (and better results) as compared to batch VI



SVI vs Batch VI on a nonparametric Bayesian Topic Model (Hierarchical Dirichlet Process)

- Learning rate (κ parameter) and minibatch size is also important (see Hoffman et al for details)

† Stochastic Variational Inference (Hoffman et al, 2013)

VI for Non-conjugate Models

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q
- The ELBO and its derivatives can be difficult to compute for non-conjugate models

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q
- The ELBO and its derivatives can be difficult to compute for non-conjugate models
- A common approach is to replace each difficult terms by a tight lower bound

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q
- The ELBO and its derivatives can be difficult to compute for non-conjugate models
- A common approach is to replace each difficult terms by a tight lower bound. Some examples:
- Assuming $q(a, b) = \prod_i q(a_i)q(b_i)$, the expectation below can be replaced by a lower bound

$$\mathbb{E}_q\left[\log \sum_i a_i b_i\right]$$

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q
- The ELBO and its derivatives can be difficult to compute for non-conjugate models
- A common approach is to replace each difficult terms by a tight lower bound. Some examples:
- Assuming $q(a, b) = \prod_i q(a_i)q(b_i)$, the expectation below can be replaced by a lower bound

$$\mathbb{E}_q\left[\log \sum_i a_i b_i\right] = \mathbb{E}_q\left[\log \sum_i p_i \frac{a_i b_i}{p_i}\right]$$

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q
- The ELBO and its derivatives can be difficult to compute for non-conjugate models
- A common approach is to replace each difficult terms by a tight lower bound. Some examples:
- Assuming $q(a, b) = \prod_i q(a_i)q(b_i)$, the expectation below can be replaced by a lower bound

$$\mathbb{E}_q\left[\log \sum_i a_i b_i\right] = \mathbb{E}_q\left[\log \sum_i p_i \frac{a_i b_i}{p_i}\right] \geq \mathbb{E}_q\left[\underbrace{\sum_i p_i \log \frac{a_i b_i}{p_i}}_{\text{via Jensen's inequality}}\right]$$

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q
- The ELBO and its derivatives can be difficult to compute for non-conjugate models
- A common approach is to replace each difficult terms by a tight lower bound. Some examples:
- Assuming $q(a, b) = \prod_i q(a_i)q(b_i)$, the expectation below can be replaced by a lower bound

$$\mathbb{E}_q\left[\log \sum_i a_i b_i\right] = \mathbb{E}_q\left[\log \sum_i p_i \frac{a_i b_i}{p_i}\right] \geq \mathbb{E}_q\left[\underbrace{\sum_i p_i \log \frac{a_i b_i}{p_i}}_{\text{via Jensen's inequality}}\right] = \sum_i p_i \mathbb{E}_q[\log a_i + \log b_i] - \sum_i p_i \log p_i$$

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q
- The ELBO and its derivatives can be difficult to compute for non-conjugate models
- A common approach is to replace each difficult terms by a tight lower bound. Some examples:
- Assuming $q(a, b) = \prod_i q(a_i)q(b_i)$, the expectation below can be replaced by a lower bound

$$\mathbb{E}_q\left[\log \sum_i a_i b_i\right] = \mathbb{E}_q\left[\log \sum_i p_i \frac{a_i b_i}{p_i}\right] \geq \mathbb{E}_q\left[\underbrace{\sum_i p_i \log \frac{a_i b_i}{p_i}}_{\text{via Jensen's inequality}}\right] = \sum_i p_i \mathbb{E}_q[\log a_i + \log b_i] - \sum_i p_i \log p_i$$

where p_i is a variable (depends on a_i and b_i) that we need to optimize

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q
- The ELBO and its derivatives can be difficult to compute for non-conjugate models
- A common approach is to replace each difficult terms by a tight lower bound. Some examples:
- Assuming $q(a, b) = \prod_i q(a_i)q(b_i)$, the expectation below can be replaced by a lower bound

$$\mathbb{E}_q\left[\log \sum_i a_i b_i\right] = \mathbb{E}_q\left[\log \sum_i p_i \frac{a_i b_i}{p_i}\right] \geq \mathbb{E}_q\left[\underbrace{\sum_i p_i \log \frac{a_i b_i}{p_i}}_{\text{via Jensen's inequality}}\right] = \sum_i p_i \mathbb{E}_q[\log a_i + \log b_i] - \sum_i p_i \log p_i$$

where p_i is a variable (depends on a_i and b_i) that we need to optimize. Expectations above easy to compute

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q
- The ELBO and its derivatives can be difficult to compute for non-conjugate models
- A common approach is to replace each difficult terms by a tight lower bound. Some examples:
- Assuming $q(a, b) = \prod_i q(a_i)q(b_i)$, the expectation below can be replaced by a lower bound

$$\mathbb{E}_q\left[\log \sum_i a_i b_i\right] = \mathbb{E}_q\left[\log \sum_i p_i \frac{a_i b_i}{p_i}\right] \geq \mathbb{E}_q\left[\underbrace{\sum_i p_i \log \frac{a_i b_i}{p_i}}_{\text{via Jensen's inequality}}\right] = \sum_i p_i \mathbb{E}_q[\log a_i + \log b_i] - \sum_i p_i \log p_i$$

  where p_i is a variable (depends on a_i and b_i) that we need to optimize. Expectations above easy to compute

- For models with logistic likelihood, we use the following (trick by Jaakkola and Jordan, 2000)

$$-\mathbb{E}_q[\log(1 + \exp(-y_n \boldsymbol{w}^\top \boldsymbol{x}_n))] \geq \log \sigma(\xi_n) + \mathbb{E}_q\left[\frac{1}{2}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \xi_n) - \lambda(\xi_n)(\boldsymbol{w}^\top \boldsymbol{x}_n \boldsymbol{x}_n^\top \boldsymbol{w} - \xi_n^2)\right]$$

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X},\mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q
- The ELBO and its derivatives can be difficult to compute for non-conjugate models
- A common approach is to replace each difficult terms by a tight lower bound. Some examples:
- Assuming $q(a,b) = \prod_i q(a_i)q(b_i)$, the expectation below can be replaced by a lower bound

$$\mathbb{E}_q\left[\log \sum_i a_i b_i\right] = \mathbb{E}_q\left[\log \sum_i p_i \frac{a_i b_i}{p_i}\right] \geq \mathbb{E}_q\left[\underbrace{\sum_i p_i \log \frac{a_i b_i}{p_i}}_{\text{via Jensen's inequality}}\right] = \sum_i p_i \mathbb{E}_q[\log a_i + \log b_i] - \sum_i p_i \log p_i$$

where p_i is a variable (depends on a_i and b_i) that we need to optimize. Expectations above easy to compute

- For models with logistic likelihood, we use the following (trick by Jaakkola and Jordan, 2000)

$$-\mathbb{E}_q[\log(1+\exp(-y_n \boldsymbol{w}^\top \boldsymbol{x}_n))] \geq \log \sigma(\xi_n) + \mathbb{E}_q\left[\frac{1}{2}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \xi_n) - \lambda(\xi_n)(\boldsymbol{w}^\top \boldsymbol{x}_n \boldsymbol{x}_n^\top \boldsymbol{w} - \xi_n^2)\right]$$

where ξ_n is a variable to be optimized and $\lambda(\xi_n) = \frac{1}{2\xi_n}[\sigma(\xi_n) - 0.5]$

Some Model-Specific Tricks

- ELBO $\mathcal{L}(q) = \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z})] - \mathbb{E}_q[\log q(\mathbf{Z})]$ requires computing expectations w.r.t. var. dist. q
- The ELBO and its derivatives can be difficult to compute for non-conjugate models
- A common approach is to replace each difficult terms by a tight lower bound. Some examples:
- Assuming $q(a,b) = \prod_i q(a_i)q(b_i)$, the expectation below can be replaced by a lower bound

$$\mathbb{E}_q\left[\log \sum_i a_i b_i\right] = \mathbb{E}_q\left[\log \sum_i p_i \frac{a_i b_i}{p_i}\right] \geq \mathbb{E}_q\left[\underbrace{\sum_i p_i \log \frac{a_i b_i}{p_i}}_{\text{via Jensen's inequality}}\right] = \sum_i p_i \mathbb{E}_q[\log a_i + \log b_i] - \sum_i p_i \log p_i$$

where p_i is a variable (depends on a_i and b_i) that we need to optimize. Expectations above easy to compute

- For models with logistic likelihood, we use the following (trick by Jaakkola and Jordan, 2000)

$$-\mathbb{E}_q[\log(1 + \exp(-y_n \boldsymbol{w}^\top \boldsymbol{x}_n))] \geq \log \sigma(\xi_n) + \mathbb{E}_q\left[\frac{1}{2}(y_n \boldsymbol{w}^\top \boldsymbol{x}_n - \xi_n) - \lambda(\xi_n)(\boldsymbol{w}^\top \boldsymbol{x}_n \boldsymbol{x}_n^\top \boldsymbol{w} - \xi_n^2)\right]$$

where ξ_n is a variable to be optimized and $\lambda(\xi_n) = \frac{1}{2\xi_n}[\sigma(\xi_n) - 0.5]$. Expectations above easy to compute

Black-box Variational Inference (BBVI)

- Black-box Variational Inference (BBVI) approximates ELBO derivatives using Monte-Carlo

*Black Box Variational Inference - Ranganath et al (2014)

Black-box Variational Inference (BBVI)

- Black-box Variational Inference (BBVI) approximates ELBO derivatives using Monte-Carlo
- Uses the following identity for the ELBO's derivative

$$\nabla_\phi \mathcal{L}(q) = \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)]$$

*Black Box Variational Inference - Ranganath et al (2014)

Black-box Variational Inference (BBVI)

- Black-box Variational Inference (BBVI) approximates ELBO derivatives using Monte-Carlo
- Uses the following identity for the ELBO's derivative

$$
\begin{aligned}
\nabla_{\phi}\mathcal{L}(q) &= \nabla_{\phi}\mathbb{E}_q[\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)] \\
&= \mathbb{E}_q[\nabla_{\phi}\log q(\mathbf{Z}|\phi)(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] \quad \text{(proof on next slide)}
\end{aligned}
$$

*Black Box Variational Inference - Ranganath et al (2014)

Black-box Variational Inference (BBVI)

- Black-box Variational Inference (BBVI) approximates ELBO derivatives using Monte-Carlo
- Uses the following identity for the ELBO's derivative

$$\begin{aligned}\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)] \\ &= \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] \quad \text{(proof on next slide)}\end{aligned}$$

- Thus ELBO gradient can be written solely in terms of expectation of gradient of $\log q(\mathbf{Z}|\phi)$

*Black Box Variational Inference - Ranganath et al (2014)

Black-box Variational Inference (BBVI)

- Black-box Variational Inference (BBVI) approximates ELBO derivatives using Monte-Carlo
- Uses the following identity for the ELBO's derivative

$$\begin{aligned}\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)] \\ &= \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi))] \quad \text{(proof on next slide)}\end{aligned}$$

- Thus ELBO gradient can be written solely in terms of expectation of gradient of log $q(\mathbf{Z}|\phi)$
  - Required gradients don't depend on the model. Only on the chosen variational distribution

*Black Box Variational Inference - Ranganath et al (2014)

Black-box Variational Inference (BBVI)

- Black-box Variational Inference (BBVI) approximates ELBO derivatives using Monte-Carlo
- Uses the following identity for the ELBO's derivative

$$\begin{aligned}\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)] \\ &= \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi))] \quad \text{(proof on next slide)}\end{aligned}$$

- Thus ELBO gradient can be written solely in terms of expectation of gradient of log $q(\mathbf{Z}|\phi)$
  - Required gradients don't depend on the model. Only on the chosen variational distribution
  - That's why this approach is called "black-box"

*Black Box Variational Inference - Ranganath et al (2014)

Black-box Variational Inference (BBVI)

- Black-box Variational Inference (BBVI) approximates ELBO derivatives using Monte-Carlo
- Uses the following identity for the ELBO's derivative

$$\begin{aligned}\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)] \\ &= \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi))] \quad \text{(proof on next slide)}\end{aligned}$$

- Thus ELBO gradient can be written solely in terms of expectation of gradient of log $q(\mathbf{Z}|\phi)$
  - Required gradients don't depend on the model. Only on the chosen variational distribution
  - That's why this approach is called "black-box"
- Given S samples $\{\mathbf{Z}_s\}_{s=1}^S$ from $q(\mathbf{Z}|\phi)$, we can get (noisy) gradient $\nabla_\phi \mathcal{L}(q)$ as follows

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S} \sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

* Black Box Variational Inference - Ranganath et al (2014)

Black-box Variational Inference (BBVI)

- Black-box Variational Inference (BBVI) approximates ELBO derivatives using Monte-Carlo
- Uses the following identity for the ELBO's derivative

$$\begin{aligned}\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \mathbb{E}_q[\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)] \\ &= \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi))] \quad \text{(proof on next slide)}\end{aligned}$$

- Thus ELBO gradient can be written solely in terms of expectation of gradient of log $q(\mathbf{Z}|\phi)$
  - Required gradients don't depend on the model. Only on the chosen variational distribution
  - That's why this approach is called "black-box"
- Given S samples $\{\mathbf{Z}_s\}_{s=1}^S$ from $q(\mathbf{Z}|\phi)$, we can get (noisy) gradient $\nabla_\phi \mathcal{L}(q)$ as follows

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

- Above is also called the "score function" based gradient (also REINFORCE method)

* Black Box Variational Inference - Ranganath et al (2014)

Proof of BBVI Identity

- The ELBO gradient can be written as

$$\nabla_{\phi}\mathcal{L}(q) \quad = \quad \nabla_{\phi}\int(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))q(\mathbf{Z}|\phi)d\mathbf{Z}$$

Proof of BBVI Identity

- The ELBO gradient can be written as

$$
\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X}, \mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi)] d\mathbf{Z} \qquad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem})
\end{aligned}
$$

Proof of BBVI Identity

- The ELBO gradient can be written as

$$
\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi)] d\mathbf{Z} \qquad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem}) \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] q(\mathbf{Z}|\phi) + \nabla_\phi q(\mathbf{Z}|\phi) [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z}
\end{aligned}
$$

Proof of BBVI Identity

- The ELBO gradient can be written as

$$
\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi)] d\mathbf{Z} \qquad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem}) \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] q(\mathbf{Z}|\phi) + \nabla_\phi q(\mathbf{Z}|\phi) [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z} \\
&= \mathbb{E}_q[-\nabla_\phi \log q(\mathbf{Z}|\phi)] + \int \nabla_\phi q(\mathbf{Z}|\phi) [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z}
\end{aligned}
$$

Proof of BBVI Identity

- The ELBO gradient can be written as

$$\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))q(\mathbf{Z}|\phi)d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))q(\mathbf{Z}|\phi)]d\mathbf{Z} \quad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem}) \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]q(\mathbf{Z}|\phi) + \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]d\mathbf{Z} \\
&= \mathbb{E}_q[-\nabla_\phi \log q(\mathbf{Z}|\phi)] + \int \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]d\mathbf{Z}
\end{aligned}$$

- Note that $\mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)] = \mathbb{E}_q\left[\frac{\nabla_\phi q(\mathbf{Z}|\phi)}{q(\mathbf{Z}|\phi)}\right]$

Proof of BBVI Identity

- The ELBO gradient can be written as

$$
\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi)] d\mathbf{Z} \quad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem}) \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] q(\mathbf{Z}|\phi) + \nabla_\phi q(\mathbf{Z}|\phi) [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z} \\
&= \mathbb{E}_q[-\nabla_\phi \log q(\mathbf{Z}|\phi)] + \int \nabla_\phi q(\mathbf{Z}|\phi) [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z}
\end{aligned}
$$

- Note that $\mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)] = \mathbb{E}_q\left[\frac{\nabla_\phi q(\mathbf{Z}|\phi)}{q(\mathbf{Z}|\phi)}\right] = \int \nabla_\phi q(\mathbf{Z}|\phi) d\mathbf{Z}$

Proof of BBVI Identity

- The ELBO gradient can be written as

$$
\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi)] d\mathbf{Z} \qquad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem}) \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] q(\mathbf{Z}|\phi) + \nabla_\phi q(\mathbf{Z}|\phi) [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z} \\
&= \mathbb{E}_q[-\nabla_\phi \log q(\mathbf{Z}|\phi)] + \int \nabla_\phi q(\mathbf{Z}|\phi) [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z}
\end{aligned}
$$

- Note that $\mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)] = \mathbb{E}_q\left[\frac{\nabla_\phi q(\mathbf{Z}|\phi)}{q(\mathbf{Z}|\phi)}\right] = \int \nabla_\phi q(\mathbf{Z}|\phi) d\mathbf{Z} = \nabla_\phi \int q(\mathbf{Z}|\phi) d\mathbf{Z}$

Proof of BBVI Identity

- The ELBO gradient can be written as

$$
\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi)] d\mathbf{Z} \qquad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem}) \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] q(\mathbf{Z}|\phi) + \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z} \\
&= \mathbb{E}_q[-\nabla_\phi \log q(\mathbf{Z}|\phi)] + \int \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z}
\end{aligned}
$$

- Note that $\mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)] = \mathbb{E}_q\left[\frac{\nabla_\phi q(\mathbf{Z}|\phi)}{q(\mathbf{Z}|\phi)}\right] = \int \nabla_\phi q(\mathbf{Z}|\phi) d\mathbf{Z} = \nabla_\phi \int q(\mathbf{Z}|\phi) d\mathbf{Z} = \nabla_\phi 1 = 0$

Proof of BBVI Identity

- The ELBO gradient can be written as

$$\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))q(\mathbf{Z}|\phi)d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))q(\mathbf{Z}|\phi)]d\mathbf{Z} \qquad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem}) \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]q(\mathbf{Z}|\phi) + \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]d\mathbf{Z} \\
&= \mathbb{E}_q[-\nabla_\phi \log q(\mathbf{Z}|\phi)] + \int \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]d\mathbf{Z}
\end{aligned}$$

- Note that $\mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)] = \mathbb{E}_q\left[\frac{\nabla_\phi q(\mathbf{Z}|\phi)}{q(\mathbf{Z}|\phi)}\right] = \int \nabla_\phi q(\mathbf{Z}|\phi)d\mathbf{Z} = \nabla_\phi \int q(\mathbf{Z}|\phi)d\mathbf{Z} = \nabla_\phi 1 = 0$
- Also note that $\nabla_\phi q(\mathbf{Z}|\phi) = \nabla_\phi[\log q(\mathbf{Z}|\phi)]q(\mathbf{Z}|\phi)$

Proof of BBVI Identity

- The ELBO gradient can be written as

$$
\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi)] d\mathbf{Z} \quad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem}) \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] q(\mathbf{Z}|\phi) + \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z} \\
&= \mathbb{E}_q[-\nabla_\phi \log q(\mathbf{Z}|\phi)] + \int \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z}
\end{aligned}
$$

- Note that $\mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)] = \mathbb{E}_q\left[\frac{\nabla_\phi q(\mathbf{Z}|\phi)}{q(\mathbf{Z}|\phi)}\right] = \int \nabla_\phi q(\mathbf{Z}|\phi) d\mathbf{Z} = \nabla_\phi \int q(\mathbf{Z}|\phi) d\mathbf{Z} = \nabla_\phi 1 = 0$

- Also note that $\nabla_\phi q(\mathbf{Z}|\phi) = \nabla_\phi [\log q(\mathbf{Z}|\phi)] q(\mathbf{Z}|\phi)$, using which

$$
\int \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z} = \int \nabla_\phi \log q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] q(\mathbf{Z}|\phi) d\mathbf{Z}
$$

Proof of BBVI Identity

- The ELBO gradient can be written as

$$\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi)] d\mathbf{Z} \qquad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem}) \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] q(\mathbf{Z}|\phi) + \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z} \\
&= \mathbb{E}_q[-\nabla_\phi \log q(\mathbf{Z}|\phi)] + \int \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z}
\end{aligned}$$

- Note that $\mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)] = \mathbb{E}_q\left[\frac{\nabla_\phi q(\mathbf{Z}|\phi)}{q(\mathbf{Z}|\phi)}\right] = \int \nabla_\phi q(\mathbf{Z}|\phi) d\mathbf{Z} = \nabla_\phi \int q(\mathbf{Z}|\phi) d\mathbf{Z} = \nabla_\phi 1 = 0$
- Also note that $\nabla_\phi q(\mathbf{Z}|\phi) = \nabla_\phi [\log q(\mathbf{Z}|\phi)] q(\mathbf{Z}|\phi)$, using which

$$\begin{aligned}
\int \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z} &= \int \nabla_\phi \log q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]
\end{aligned}$$

Proof of BBVI Identity

- The ELBO gradient can be written as

$$\begin{aligned}
\nabla_\phi \mathcal{L}(q) &= \nabla_\phi \int (\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi)) q(\mathbf{Z}|\phi)] d\mathbf{Z} \quad (\nabla \text{ and } \int \text{ interchangeable; dominated convergence theorem}) \\
&= \int \nabla_\phi [(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] q(\mathbf{Z}|\phi) + \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z} \\
&= \mathbb{E}_q[-\nabla_\phi \log q(\mathbf{Z}|\phi)] + \int \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z}
\end{aligned}$$

- Note that $\mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)] = \mathbb{E}_q\left[\frac{\nabla_\phi q(\mathbf{Z}|\phi)}{q(\mathbf{Z}|\phi)}\right] = \int \nabla_\phi q(\mathbf{Z}|\phi) d\mathbf{Z} = \nabla_\phi \int q(\mathbf{Z}|\phi) d\mathbf{Z} = \nabla_\phi 1 = 0$
- Also note that $\nabla_\phi q(\mathbf{Z}|\phi) = \nabla_\phi[\log q(\mathbf{Z}|\phi)] q(\mathbf{Z}|\phi)$, using which

$$\begin{aligned}
\int \nabla_\phi q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] d\mathbf{Z} &= \int \nabla_\phi \log q(\mathbf{Z}|\phi)[(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))] q(\mathbf{Z}|\phi) d\mathbf{Z} \\
&= \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]
\end{aligned}$$

- Therefore $\nabla_\phi \mathcal{L}(q) = \mathbb{E}_q[\nabla_\phi \log q(\mathbf{Z}|\phi)(\log p(\mathbf{X},\mathbf{Z}) - \log q(\mathbf{Z}|\phi))]$

Benefits of BBVI

- Recall that BBVI approximates the ELBO gradients by the Monte Carlo expectations

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

Benefits of BBVI

- Recall that BBVI approximates the ELBO gradients by the Monte Carlo expectations

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X},\mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

- Enables applying VB inference for a wide variety of probabilistic models

Benefits of BBVI

- Recall that BBVI approximates the ELBO gradients by the Monte Carlo expectations

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

- Enables applying VB inference for a wide variety of probabilistic models
- Can also work with small minibatches of data rather than full data

Benefits of BBVI

- Recall that BBVI approximates the ELBO gradients by the Monte Carlo expectations

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

- Enables applying VB inference for a wide variety of probabilistic models
- Can also work with small minibatches of data rather than full data
- BBVI has very few requirements

Benefits of BBVI

- Recall that BBVI approximates the ELBO gradients by the Monte Carlo expectations

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

- Enables applying VB inference for a wide variety of probabilistic models
- Can also work with small minibatches of data rather than full data
- BBVI has very few requirements
  - Should be able to sample from $q(\mathbf{Z}|\phi)$

Benefits of BBVI

- Recall that BBVI approximates the ELBO gradients by the Monte Carlo expectations

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

- Enables applying VB inference for a wide variety of probabilistic models
- Can also work with small minibatches of data rather than full data
- BBVI has very few requirements
  - Should be able to sample from $q(\mathbf{Z}|\phi)$
  - Should be able to compute $\nabla_\phi \log q(\mathbf{Z}|\phi)$ (automatic differentiation methods exist!)

Benefits of BBVI

- Recall that BBVI approximates the ELBO gradients by the Monte Carlo expectations

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

- Enables applying VB inference for a wide variety of probabilistic models
- Can also work with small minibatches of data rather than full data
- BBVI has very few requirements
  - Should be able to sample from $q(\mathbf{Z}|\phi)$
  - Should be able to compute $\nabla_\phi \log q(\mathbf{Z}|\phi)$ (automatic differentiation methods exist!)
  - Should be able to evaluate $p(\mathbf{X}, \mathbf{Z})$ and $\log q(\mathbf{Z}|\phi)$

Benefits of BBVI

- Recall that BBVI approximates the ELBO gradients by the Monte Carlo expectations

$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

- Enables applying VB inference for a wide variety of probabilistic models
- Can also work with small minibatches of data rather than full data
- BBVI has very few requirements
  - Should be able to sample from $q(\mathbf{Z}|\phi)$
  - Should be able to compute $\nabla_\phi \log q(\mathbf{Z}|\phi)$ (automatic differentiation methods exist!)
  - Should be able to evaluate $p(\mathbf{X}, \mathbf{Z})$ and $\log q(\mathbf{Z}|\phi)$
- Some tricks needed to control the variance in the Monte Carlo estimate of the ELBO gradient (if interested in the details, please refer to the BBVI paper)

Reparametrization Trick

- Another Monte-Carlo approx. of ELBO grad (with often lower variance than BBVI based grad)

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick

- Another Monte-Carlo approx. of ELBO grad (with often lower variance than BBVI based grad)
- In general, suppose we want to compute ELBO's gradient $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})]$

*Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick

- Another Monte-Carlo approx. of ELBO grad (with often lower variance than BBVI based grad)
- In general, suppose we want to compute ELBO's gradient $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})]$
- Assume a deterministic transformation $\mathbf{Z} = g(\epsilon, \phi)$ with $\epsilon \sim p(\epsilon)$, and $p(\epsilon)$ doesn't depend on ϕ

*Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick

- Another Monte-Carlo approx. of ELBO grad (with often lower variance than BBVI based grad)
- In general, suppose we want to compute ELBO's gradient $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})]$
- Assume a deterministic transformation $\mathbf{Z} = g(\epsilon, \phi)$ with $\epsilon \sim p(\epsilon)$, and $p(\epsilon)$ doesn't depend on ϕ
- With this reparametrization, the ELBO's gradient can be written as

$$\nabla_\phi \mathbb{E}_{p(\epsilon)}[\log p(\mathbf{X}, g(\epsilon, \phi)) - \log q_\phi(g(\epsilon, \phi))]$$

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick

- Another Monte-Carlo approx. of ELBO grad (with often lower variance than BBVI based grad)
- In general, suppose we want to compute ELBO's gradient $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})]$
- Assume a deterministic transformation $\mathbf{Z} = g(\epsilon, \phi)$ with $\epsilon \sim p(\epsilon)$, and $p(\epsilon)$ doesn't depend on ϕ
- With this reparametrization, the ELBO's gradient can be written as

$$\nabla_\phi \mathbb{E}_{p(\epsilon)}[\log p(\mathbf{X}, g(\epsilon, \phi)) - \log q_\phi(g(\epsilon, \phi))] = \mathbb{E}_{p(\epsilon)} \nabla_\phi [\log p(\mathbf{X}, g(\epsilon, \phi)) - \log q_\phi(g(\epsilon, \phi))]$$

*Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick

- Another Monte-Carlo approx. of ELBO grad (with often lower variance than BBVI based grad)
- In general, suppose we want to compute ELBO's gradient $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})]$
- Assume a deterministic transformation $\mathbf{Z} = g(\epsilon, \phi)$ with $\epsilon \sim p(\epsilon)$, and $p(\epsilon)$ doesn't depend on ϕ
- With this reparametrization, the ELBO's gradient can be written as

$$\nabla_\phi \mathbb{E}_{p(\epsilon)}[\log p(\mathbf{X}, g(\epsilon, \phi)) - \log q_\phi(g(\epsilon, \phi))] = \mathbb{E}_{p(\epsilon)} \nabla_\phi [\log p(\mathbf{X}, g(\epsilon, \phi)) - \log q_\phi(g(\epsilon, \phi))]$$

- LHS true due to Law of Unconscious Statistician

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick

- Another Monte-Carlo approx. of ELBO grad (with often lower variance than BBVI based grad)
- In general, suppose we want to compute ELBO's gradient $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})]$
- Assume a deterministic transformation $\mathbf{Z} = g(\epsilon, \phi)$ with $\epsilon \sim p(\epsilon)$, and $p(\epsilon)$ doesn't depend on ϕ
- With this reparametrization, the ELBO's gradient can be written as

$$\nabla_\phi \mathbb{E}_{p(\epsilon)}[\log p(\mathbf{X}, g(\epsilon, \phi)) - \log q_\phi(g(\epsilon, \phi))] = \mathbb{E}_{p(\epsilon)} \nabla_\phi [\log p(\mathbf{X}, g(\epsilon, \phi)) - \log q_\phi(g(\epsilon, \phi))]$$

- LHS true due to Law of Unconscious Statistician
- Could interchange expect. and grad. on RHS since $p(\epsilon)$ doesn't depend on ϕ

*Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick

- Another Monte-Carlo approx. of ELBO grad (with often lower variance than BBVI based grad)
- In general, suppose we want to compute ELBO's gradient $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})]$
- Assume a deterministic transformation $\mathbf{Z} = g(\epsilon,\phi)$ with $\epsilon \sim p(\epsilon)$, and $p(\epsilon)$ doesn't depend on ϕ
- With this reparametrization, the ELBO's gradient can be written as

$$\nabla_\phi \mathbb{E}_{p(\epsilon)}[\log p(\mathbf{X}, g(\epsilon,\phi)) - \log q_\phi(g(\epsilon,\phi))] = \mathbb{E}_{p(\epsilon)} \nabla_\phi [\log p(\mathbf{X}, g(\epsilon,\phi)) - \log q_\phi(g(\epsilon,\phi))]$$

- LHS true due to Law of Unconscious Statistician
- Could interchange expect. and grad. on RHS since $p(\epsilon)$ doesn't depend on ϕ
- Given S i.i.d. random samples $\{\epsilon_s\}_{s=1}^S$ from $p(\epsilon)$, we can compute a Monte-Carlo approx, so

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})] \approx \frac{1}{S}\sum_{s=1}^{S}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon_s,\phi)) - \nabla_\phi \log q_\phi(g(\epsilon_s,\phi))]$$

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick

- Another Monte-Carlo approx. of ELBO grad (with often lower variance than BBVI based grad)
- In general, suppose we want to compute ELBO's gradient $\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})]$
- Assume a deterministic transformation $\mathbf{Z} = g(\epsilon, \phi)$ with $\epsilon \sim p(\epsilon)$, and $p(\epsilon)$ doesn't depend on ϕ
- With this reparametrization, the ELBO's gradient can be written as

$$\nabla_\phi \mathbb{E}_{p(\epsilon)}[\log p(\mathbf{X}, g(\epsilon, \phi)) - \log q_\phi(g(\epsilon, \phi))] = \mathbb{E}_{p(\epsilon)} \nabla_\phi [\log p(\mathbf{X}, g(\epsilon, \phi)) - \log q_\phi(g(\epsilon, \phi))]$$

- LHS true due to Law of Unconscious Statistician
- Could interchange expect. and grad. on RHS since $p(\epsilon)$ doesn't depend on ϕ
- Given S i.i.d. random samples $\{\epsilon_s\}_{s=1}^S$ from $p(\epsilon)$, we can compute a Monte-Carlo approx, so

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})] \approx \frac{1}{S} \sum_{s=1}^{S} [\nabla_\phi \log p(\mathbf{X}, g(\epsilon_s, \phi)) - \nabla_\phi \log q_\phi(g(\epsilon_s, \phi))]$$

- Such gradients are called pathwise gradients (we took a "path" from ϵ to $\mathbf{Z}$)

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \mu + \mathbf{L}\boldsymbol{v}$

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \mu + \mathbf{L}\boldsymbol{v}$ where $\mathbf{L} = \text{chol}(\Sigma)$ and $\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{I})$

*Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \mu + \mathbf{L}\boldsymbol{v}$ where $\mathbf{L} = \text{chol}(\Sigma)$ and $\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{I})$, and write

$$\nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})]$$

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \mu + \mathbf{L}\boldsymbol{v}$ where $\mathbf{L} = \text{chol}(\Sigma)$ and $\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{I})$, and write

$$\nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] = \nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[f(\mu + \mathbf{L}\boldsymbol{v})]$$

*Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \mu + \mathbf{L}\boldsymbol{v}$ where $\mathbf{L} = \text{chol}(\Sigma)$ and $\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{I})$, and write

$$\nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] = \nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[f(\mu + \mathbf{L}\boldsymbol{v})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_{\mu,\mathbf{L}} f(\mu + \mathbf{L}\boldsymbol{v})]$$

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \mu + \mathbf{L}\boldsymbol{v}$ where $\mathbf{L} = \text{chol}(\Sigma)$ and $\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{I})$, and write

$$\nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] = \nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[f(\mu + \mathbf{L}\boldsymbol{v})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_{\mu,\mathbf{L}} f(\mu + \mathbf{L}\boldsymbol{v})]$$

- Now easy to take derivatives w.r.t. variational params $\mu, \mathbf{L}$ using Monte Carlo sampling

---

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \mu + \mathbf{L}\boldsymbol{v}$ where $\mathbf{L} = \text{chol}(\Sigma)$ and $\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{I})$, and write

$$\nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] = \nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[f(\mu + \mathbf{L}\boldsymbol{v})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_{\mu,\mathbf{L}} f(\mu + \mathbf{L}\boldsymbol{v})]$$

- Now easy to take derivatives w.r.t. variational params $\mu, \mathbf{L}$ using Monte Carlo sampling
- In practice, even one random sample $\boldsymbol{v}_s \sim \mathcal{N}(\boldsymbol{v}|0, \mathbf{I})$ suffices*

*Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \mu + \mathbf{L}\boldsymbol{v}$ where $\mathbf{L} = \text{chol}(\Sigma)$ and $\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{I})$, and write

$$\nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] = \nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[f(\mu + \mathbf{L}\boldsymbol{v})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_{\mu,\mathbf{L}} f(\mu + \mathbf{L}\boldsymbol{v})]$$

- Now easy to take derivatives w.r.t. variational params $\mu, \mathbf{L}$ using Monte Carlo sampling
- In practice, even one random sample $\boldsymbol{v}_s \sim \mathcal{N}(\boldsymbol{v}|0, \mathbf{I})$ suffices*. So the above gradients will be

$$\nabla_\mu \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_\mu f(\mu + \mathbf{L}\boldsymbol{v})] \approx \nabla_\mu f(\mu + \mathbf{L}\boldsymbol{v}_s)$$

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \mu + \mathbf{L}\boldsymbol{v}$ where $\mathbf{L} = \text{chol}(\Sigma)$ and $\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{I})$, and write

$$\nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] = \nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[f(\mu + \mathbf{L}\boldsymbol{v})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_{\mu,\mathbf{L}} f(\mu + \mathbf{L}\boldsymbol{v})]$$

- Now easy to take derivatives w.r.t. variational params $\mu, \mathbf{L}$ using Monte Carlo sampling
- In practice, even one random sample $\boldsymbol{v}_s \sim \mathcal{N}(\boldsymbol{v}|0, \mathbf{I})$ suffices*. So the above gradients will be

$$\begin{aligned}
\nabla_\mu \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] &= \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_\mu f(\mu + \mathbf{L}\boldsymbol{v})] \approx \nabla_\mu f(\mu + \mathbf{L}\boldsymbol{v}_s) \\
\nabla_\mathbf{L} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] &= \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_\mathbf{L} f(\mu + \mathbf{L}\boldsymbol{v})] \approx \nabla_\mathbf{L} f(\mu + \mathbf{L}\boldsymbol{v}_s)
\end{aligned}$$

* Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \mu + \mathbf{L}\boldsymbol{v}$ where $\mathbf{L} = \text{chol}(\Sigma)$ and $\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{I})$, and write

$$\nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] = \nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[f(\mu + \mathbf{L}\boldsymbol{v})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_{\mu,\mathbf{L}} f(\mu + \mathbf{L}\boldsymbol{v})]$$

- Now easy to take derivatives w.r.t. variational params $\mu, \mathbf{L}$ using Monte Carlo sampling
- In practice, even one random sample $\boldsymbol{v}_s \sim \mathcal{N}(\boldsymbol{v}|0, \mathbf{I})$ suffices*. So the above gradients will be

$$\begin{aligned}
\nabla_\mu \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] &= \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_\mu f(\mu + \mathbf{L}\boldsymbol{v})] \approx \nabla_\mu f(\mu + \mathbf{L}\boldsymbol{v}_s) \\
\nabla_{\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] &= \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_{\mathbf{L}} f(\mu + \mathbf{L}\boldsymbol{v})] \approx \nabla_{\mathbf{L}} f(\mu + \mathbf{L}\boldsymbol{v}_s)
\end{aligned}$$

.. the above just requires being able to take derivatives of $f(\boldsymbol{w})$ w.r.t. $\boldsymbol{w}$

*Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: An Example

- Suppose our variational distribution is $q_\phi(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu, \Sigma)$, so $\phi = \{\mu, \Sigma\}$
- Suppose our ELBO has a difficult term $\mathbb{E}_q[f(\boldsymbol{w})]$ (due to the expectation being intractable)
- We are actually interested in its gradient $\nabla_\phi \mathbb{E}_q[f(\boldsymbol{w})]$. Let's use the reparametrization trick
- Reparametrize $\boldsymbol{w}$ as $\boldsymbol{w} = \mu + \mathbf{L}\boldsymbol{v}$ where $\mathbf{L} = \text{chol}(\Sigma)$ and $\boldsymbol{v} \sim \mathcal{N}(0, \mathbf{I})$, and write

$$\nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] = \nabla_{\mu,\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[f(\mu + \mathbf{L}\boldsymbol{v})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_{\mu,\mathbf{L}} f(\mu + \mathbf{L}\boldsymbol{v})]$$

- Now easy to take derivatives w.r.t. variational params $\mu, \mathbf{L}$ using Monte Carlo sampling
- In practice, even one random sample $\boldsymbol{v}_s \sim \mathcal{N}(\boldsymbol{v}|0, \mathbf{I})$ suffices*. So the above gradients will be

$$\begin{aligned}
\nabla_\mu \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] &= \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_\mu f(\mu + \mathbf{L}\boldsymbol{v})] \approx \nabla_\mu f(\mu + \mathbf{L}\boldsymbol{v}_s) \\
\nabla_{\mathbf{L}} \mathbb{E}_{\mathcal{N}(\boldsymbol{w}|\mu,\Sigma)}[f(\boldsymbol{w})] &= \mathbb{E}_{\mathcal{N}(\boldsymbol{v}|0,\mathbf{I})}[\nabla_{\mathbf{L}} f(\mu + \mathbf{L}\boldsymbol{v})] \approx \nabla_{\mathbf{L}} f(\mu + \mathbf{L}\boldsymbol{v}_s)
\end{aligned}$$

  .. the above just requires being able to take derivatives of $f(\boldsymbol{w})$ w.r.t. $\boldsymbol{w}$
- Note: Std. reparam. trick assumes differentiability but recent work on removing this limitation

*Autoencoding Variational Bayes - Kingma and Welling (2013)

Reparametrization Trick: Some Comments

- Standard Reparametrization Trick assumes the model to be differentiable

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon, \phi)) - \nabla_\phi \log q_\phi(g(\epsilon, \phi))]$$

† Categorical Reparameterization with Gumbel-Softmax (Jang et al, 2017), * The Generalized Reparameterization Gradient (Ruiz et al, 2016), # Implicit Reparameterization Gradients (Figurnov et al, 2018), @ Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms (Naesseth et al, 2016)

Reparametrization Trick: Some Comments

- Standard Reparametrization Trick assumes the model to be differentiable

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon,\phi)) - \nabla_\phi \log q_\phi(g(\epsilon,\phi))]$$

- Note that this wasn't the case with BBVI

† Categorical Reparameterization with Gumbel-Softmax (Jang et al, 2017), * The Generalized Reparameterization Gradient (Ruiz et al, 2016), # Implicit Reparameterization Gradients (Figurnov et al, 2018), @ Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms (Naesseth et al, 2016)

Reparametrization Trick: Some Comments

- Standard Reparametrization Trick assumes the model to be differentiable

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon,\phi)) - \nabla_\phi \log q_\phi(g(\epsilon,\phi))]$$

- Note that this wasn't the case with BBVI
- Thus rep. trick often isn't applicable, e.g., when **Z** is discrete (e.g., binary, categorical, etc)

† Categorical Reparameterization with Gumbel-Softmax (Jang et al, 2017), * The Generalized Reparameterization Gradient (Ruiz et al, 2016), # Implicit Reparameterization Gradients (Figurnov et al, 2018), @ Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms (Naesseth et al, 2016)

Reparametrization Trick: Some Comments

- Standard Reparametrization Trick assumes the model to be differentiable

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon,\phi)) - \nabla_\phi \log q_\phi(g(\epsilon,\phi))]$$

- Note that this wasn't the case with BBVI
- Thus rep. trick often isn't applicable, e.g., when $\mathbf{Z}$ is discrete (e.g., binary, categorical, etc)
  - Recent work on continuous approximation of discrete variables[†]

[†] Categorical Reparameterization with Gumbel-Softmax (Jang et al, 2017), [*] The Generalized Reparameterization Gradient (Ruiz et al, 2016), [#] Implicit Reparameterization Gradients (Figurnov et al, 2018), [@] Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms (Naesseth et al, 2016)

Reparametrization Trick: Some Comments

- Standard Reparametrization Trick assumes the model to be differentiable

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon,\phi)) - \nabla_\phi \log q_\phi(g(\epsilon,\phi))]$$

- Note that this wasn't the case with BBVI
- Thus rep. trick often isn't applicable, e.g., when $\mathbf{Z}$ is discrete (e.g., binary, categorical, etc)
  - Recent work on continuous approximation of discrete variables[†]
- The transformation function g may be difficult to find for general distributions

[†] Categorical Reparameterization with Gumbel-Softmax (Jang et al, 2017), [*] The Generalized Reparameterization Gradient (Ruiz et al, 2016), [#] Implicit Reparameterization Gradients (Figurnov et al, 2018), [@] Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms (Naesseth et al, 2016)

Reparametrization Trick: Some Comments

- Standard Reparametrization Trick assumes the model to be differentiable

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon, \phi)) - \nabla_\phi \log q_\phi(g(\epsilon, \phi))]$$

- Note that this wasn't the case with BBVI
- Thus rep. trick often isn't applicable, e.g., when $\mathbf{Z}$ is discrete (e.g., binary, categorical, etc)
  - Recent work on continuous approximation of discrete variables[†]
- The transformation function g may be difficult to find for general distributions
  - Recent work on generalized reparametrizations[*]

[†] Categorical Reparameterization with Gumbel-Softmax (Jang et al, 2017), [*] The Generalized Reparameterization Gradient (Ruiz et al, 2016), [#] Implicit Reparameterization Gradients (Figurnov et al, 2018), [@] Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms (Naesseth et al, 2016)

Reparametrization Trick: Some Comments

- Standard Reparametrization Trick assumes the model to be differentiable

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon,\phi)) - \nabla_\phi \log q_\phi(g(\epsilon,\phi))]$$

- Note that this wasn't the case with BBVI
- Thus rep. trick often isn't applicable, e.g., when **Z** is discrete (e.g., binary, categorical, etc)
  - Recent work on continuous approximation of discrete variables[†]
- The transformation function g may be difficult to find for general distributions
  - Recent work on generalized reparametrizations[*]
- Also, the transformation function g needs to be invertible (difficult/expensive)

[†] Categorical Reparameterization with Gumbel-Softmax (Jang et al, 2017), [*] The Generalized Reparameterization Gradient (Ruiz et al, 2016), [#] Implicit Reparameterization Gradients (Figurnov et al, 2018), [@] Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms (Naesseth et al, 2016)

Reparametrization Trick: Some Comments

- Standard Reparametrization Trick assumes the model to be differentiable

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X}, \mathbf{Z}) - \log q_\phi(\mathbf{Z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon, \phi)) - \nabla_\phi \log q_\phi(g(\epsilon, \phi))]$$

- Note that this wasn't the case with BBVI
- Thus rep. trick often isn't applicable, e.g., when **Z** is discrete (e.g., binary, categorical, etc)
  - Recent work on continuous approximation of discrete variables[†]
- The transformation function g may be difficult to find for general distributions
  - Recent work on generalized reparametrizations[*]
- Also, the transformation function g needs to be invertible (difficult/expensive)
  - Recent work on implicit reparametrized gradients[#]

[†] Categorical Reparameterization with Gumbel-Softmax (Jang et al, 2017), [*] The Generalized Reparameterization Gradient (Ruiz et al, 2016), [#] Implicit Reparameterization Gradients (Figurnov et al, 2018), [@] Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms (Naesseth et al, 2016)

Reparametrization Trick: Some Comments

- Standard Reparametrization Trick assumes the model to be differentiable

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon,\phi)) - \nabla_\phi \log q_\phi(g(\epsilon,\phi))]$$

- Note that this wasn't the case with BBVI
- Thus rep. trick often isn't applicable, e.g., when $\mathbf{Z}$ is discrete (e.g., binary, categorical, etc)
  - Recent work on continuous approximation of discrete variables[†]
- The transformation function g may be difficult to find for general distributions
  - Recent work on generalized reparametrizations[*]
- Also, the transformation function g needs to be invertible (difficult/expensive)
  - Recent work on implicit reparametrized gradients[#]
- Also assume that we can directly draw samples from $p(\epsilon)$. If we can't then rep. trick isn't valid[@]

[†] Categorical Reparameterization with Gumbel-Softmax (Jang et al, 2017), [*] The Generalized Reparameterization Gradient (Ruiz et al, 2016), [#] Implicit Reparameterization Gradients (Figurnov et al, 2018), [@] Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms (Naesseth et al, 2016)

Reparametrization Trick: Some Comments

- Standard Reparametrization Trick assumes the model to be differentiable

$$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{Z})}[\log p(\mathbf{X},\mathbf{Z}) - \log q_\phi(\mathbf{Z})] = \mathbb{E}_{p(\epsilon)}[\nabla_\phi \log p(\mathbf{X}, g(\epsilon,\phi)) - \nabla_\phi \log q_\phi(g(\epsilon,\phi))]$$

- Note that this wasn't the case with BBVI
- Thus rep. trick often isn't applicable, e.g., when **Z** is discrete (e.g., binary, categorical, etc)
  - Recent work on continuous approximation of discrete variables[†]
- The transformation function g may be difficult to find for general distributions
  - Recent work on generalized reparametrizations[*]
- Also, the transformation function g needs to be invertible (difficult/expensive)
  - Recent work on implicit reparametrized gradients[#]
- Also assume that we can directly draw samples from $p(\epsilon)$. If we can't then rep. trick isn't valid[@]
- Very active area of research in VI right now!

[†] Categorical Reparameterization with Gumbel-Softmax (Jang et al, 2017), [*] The Generalized Reparameterization Gradient (Ruiz et al, 2016), [#] Implicit Reparameterization Gradients (Figurnov et al, 2018), [@] Reparameterization Gradients through Acceptance-Rejection Sampling Algorithms (Naesseth et al, 2016)

Automatic Differentiation Variational Inference (ADVI)

- Auto. Diff. (AD): A way to automate differentiation of functions with unconstrained variables

---

*Automatic Differentiation Variational Inference (Kucukelbir et al, 2017)

Automatic Differentiation Variational Inference (ADVI)

- Auto. Diff. (AD): A way to automate differentiation of functions with unconstrained variables
- These derivatives is all what we need to optimize the function (in our case, ELBO)

* Automatic Differentiation Variational Inference (Kucukelbir et al, 2017)

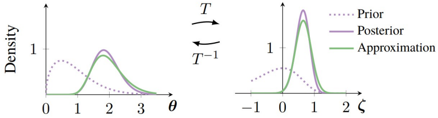
Automatic Differentiation Variational Inference (ADVI)

- Auto. Diff. (AD): A way to automate differentiation of functions with unconstrained variables
- These derivatives is all what we need to optimize the function (in our case, ELBO)
- VI is also optimization. However, often the variables are constrained, e.g.,
  - Gamma's shape and scale can only be non-negative
  - Beta's parameters can only be non-negative
  - Dirichlet's probability parameter sums to one

*Automatic Differentiation Variational Inference (Kucukelbir et al, 2017)

Automatic Differentiation Variational Inference (ADVI)

- Auto. Diff. (AD): A way to automate differentiation of functions with unconstrained variables
- These derivatives is all what we need to optimize the function (in our case, ELBO)
- VI is also optimization. However, often the variables are constrained, e.g.,
  - Gamma's shape and scale can only be non-negative
  - Beta's parameters can only be non-negative
  - Dirichlet's probability parameter sums to one
- If we can somehow transform our distributions to unconstrained ones, we can use AD for VI



(a) Latent variable space **(b)** Real coordinate space

$$T : \text{supp}(p(\boldsymbol{\theta})) \to \mathbb{R}^K$$

$$\boldsymbol{\zeta} = T(\boldsymbol{\theta})$$

$$p(\mathbf{x}, \boldsymbol{\zeta}) = p\left(\mathbf{x}, T^{-1}(\boldsymbol{\zeta})\right) \left| \det J_{T^{-1}}(\boldsymbol{\zeta}) \right|$$

Transformed density | Original density | Jacobian of inverse of T

*Automatic Differentiation Variational Inference (Kucukelbir et al, 2017)

Amortized Variational Inference

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates
    - Update local ϕ_*, update global λ, and repeat until convergence

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates
  - Update local ϕ_*, update global λ, and repeat until convergence
- Amortized VI : Learn an "inference network" or "recognition model" to directly get ϕ_n

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates
  - Update local ϕ_*, update global λ, and repeat until convergence
- Amortized VI : Learn an "inference network" or "recognition model" to directly get ϕ_n, e.g.,
  - A neural network to directly map $\boldsymbol{x}_n$ to ϕ_n

$$q(\boldsymbol{z}_n|\phi_n) \approx q(\boldsymbol{z}_n|\hat{\phi}_n) \quad \text{where} \quad \hat{\phi}_n = \text{NN}_\phi(\boldsymbol{x}_n)$$

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates
  - Update local ϕ_*, update global λ, and repeat until convergence
- Amortized VI : Learn an "inference network" or "recognition model" to directly get ϕ_n, e.g.,
  - A neural network to directly map $\boldsymbol{x}_n$ to ϕ_n

$$q(\boldsymbol{z}_n|\phi_n) \approx q(\boldsymbol{z}_n|\hat{\phi}_n) \quad \text{where} \quad \hat{\phi}_n = \text{NN}_\phi(\boldsymbol{x}_n)$$

- The inference network params ϕ can be learned along with the other global vars

Amortized Variational Inference

- Many latent variable models have one latent variable $\boldsymbol{z}_n$ for each data point $\boldsymbol{x}_n$
- VI finds the optimal ϕ_n for each $q(\boldsymbol{z}_n|\phi_n)$
- This can be expensive for large datasets (a similar issue which motivated SVI)
- Also slow at test time: Given a new $\boldsymbol{x}_*$, finding ϕ_* requires iterative updates
  - Update local ϕ_*, update global λ, and repeat until convergence
- Amortized VI : Learn an "inference network" or "recognition model" to directly get ϕ_n, e.g.,
  - A neural network to directly map $\boldsymbol{x}_n$ to ϕ_n

$$q(\boldsymbol{z}_n|\phi_n) \approx q(\boldsymbol{z}_n|\hat{\phi}_n) \quad \text{where} \quad \hat{\phi}_n = \text{NN}_\phi(\boldsymbol{x}_n)$$

- The inference network params ϕ can be learned along with the other global vars
- Popular in deep probabilistic models such as variational autoencoders, deep Gaussian Processes, etc

Structured Variational Inference

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist
  - Structured mean-field (Saul et al, 1996)

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist
  - Structured mean-field (Saul et al, 1996)
  - Hierarchical VI (Ranganath et al, 2016): Variational params $\phi_1, \ldots, \phi_M$ "tied" via a shared prior

$$q(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M | \theta) = \int \left[\prod_{m=1}^{M} q(\boldsymbol{z}_m | \phi_m) \right] p(\phi | \theta) d\phi$$

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist
  - Structured mean-field (Saul et al, 1996)
  - Hierarchical VI (Ranganath et al, 2016): Variational params $\phi_1, \ldots, \phi_M$ "tied" via a shared prior

$$q(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M | \theta) = \int \left[\prod_{m=1}^{M} q(\boldsymbol{z}_m | \phi_m) \right] p(\boldsymbol{\phi} | \theta) d\boldsymbol{\phi}$$

- To learn more expressive variational approximations, various approaches exist

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist
  - Structured mean-field (Saul et al, 1996)
  - Hierarchical VI (Ranganath et al, 2016): Variational params $\phi_1, \ldots, \phi_M$ "tied" via a shared prior

$$q(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M | \theta) = \int \left[\prod_{m=1}^{M} q(\boldsymbol{z}_m | \phi_m) \right] p(\boldsymbol{\phi} | \theta) d\boldsymbol{\phi}$$

- To learn more expressive variational approximations, various approaches exist, e.g.,
  - Boosting or mixture of simpler distributions, e.g., $q(\boldsymbol{z}) = \sum_{c=1}^{C} \rho_c q_c(\boldsymbol{z} | \phi_c)$

Structured Variational Inference

- Here "structured" may refer to anything that makes the VI approximation more expressive, e.g.,
  - Removing the independence assumption of mean-field VI
  - Learning more complex forms variational distributions
- To remove the mean-field assumption, various approaches exist
  - Structured mean-field (Saul et al, 1996)
  - Hierarchical VI (Ranganath et al, 2016): Variational params $\phi_1, \ldots, \phi_M$ "tied" via a shared prior

$$q(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M | \theta) = \int \left[\prod_{m=1}^{M} q(\boldsymbol{z}_m | \phi_m) \right] p(\boldsymbol{\phi} | \theta) d\boldsymbol{\phi}$$

- To learn more expressive variational approximations, various approaches exist, e.g.,
  - Boosting or mixture of simpler distributions, e.g., $q(\boldsymbol{z}) = \sum_{c=1}^{C} \rho_c q_c(\boldsymbol{z}|\phi_c)$
  - Normalizing flows:' Turn a simple $q(\boldsymbol{z})$ into a complex one via series of invertible transformations

Other Divergence Measures

Other Divergence Measures

- VI minimizes $KL(q||p)$ but other divergences can be minimized as well

Other Divergence Measures

- VI minimizes $KL(q||p)$ but other divergences can be minimized as well
- A general form of divergence is Renyi's α-divergence defined as

$$D_\alpha^R(p(\boldsymbol{x})||q(\boldsymbol{x})) = \frac{1}{\alpha - 1} \log \int p(\boldsymbol{x})^\alpha q(\boldsymbol{x})^{1-\alpha} d\boldsymbol{x}$$

Other Divergence Measures

- VI minimizes $KL(q||p)$ but other divergences can be minimized as well
- A general form of divergence is Renyi's α-divergence defined as

$$D_{\alpha}^{R}(p(\boldsymbol{x})||q(\boldsymbol{x})) = \frac{1}{\alpha - 1} \log \int p(\boldsymbol{x})^{\alpha} q(\boldsymbol{x})^{1-\alpha} d\boldsymbol{x}$$

- $KL(p||q)$ is a special case with $\alpha \to 1$ (can verify using L'Hopital rule of taking limits)

Other Divergence Measures

- VI minimizes $KL(q||p)$ but other divergences can be minimized as well
- A general form of divergence is Renyi's α-divergence defined as

$$D_\alpha^R(p(\boldsymbol{x})||q(\boldsymbol{x})) = \frac{1}{\alpha - 1}\log\int p(\boldsymbol{x})^\alpha q(\boldsymbol{x})^{1-\alpha} d\boldsymbol{x}$$

- $KL(p||q)$ is a special case with $\alpha \to 1$ (can verify using L'Hopital rule of taking limits)
- An even more general form of divergece is f-Divergence

$$D_f(p(\boldsymbol{x})||q(\boldsymbol{x})) = \int q(\boldsymbol{x}) f\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right) d\boldsymbol{x}$$

Other Divergence Measures

- VI minimizes $KL(q||p)$ but other divergences can be minimized as well
- A general form of divergence is Renyi's α-divergence defined as

$$D_\alpha^R(p(\boldsymbol{x})||q(\boldsymbol{x})) = \frac{1}{\alpha - 1}\log\int p(\boldsymbol{x})^\alpha q(\boldsymbol{x})^{1-\alpha}d\boldsymbol{x}$$

- $KL(p||q)$ is a special case with $\alpha \to 1$ (can verify using L'Hopital rule of taking limits)
- An even more general form of divergece is f-Divergence

$$D_f(p(\boldsymbol{x})||q(\boldsymbol{x})) = \int q(\boldsymbol{x})f\left(\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}\right)d\boldsymbol{x}$$

- Many recent inference algorithms are based on minimizing such divergences

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference
- Even mean-field for locally-conjugate models has many applications in lots of probabilistic models
  - This + SVI gives excellent scalability

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference
- Even mean-field for locally-conjugate models has many applications in lots of probabilistic models
  - This + SVI gives excellent scalability
- Stoch. opt., auto. diff., Monte-Carlo gradient of ELBO, contributed immensely to the success

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference
- Even mean-field for locally-conjugate models has many applications in lots of probabilistic models
  - This + SVI gives excellent scalability
- Stoch. opt., auto. diff., Monte-Carlo gradient of ELBO, contributed immensely to the success
- Note: Most of these ideas apply also to Variational EM

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference
- Even mean-field for locally-conjugate models has many applications in lots of probabilistic models
  - This + SVI gives excellent scalability
- Stoch. opt., auto. diff., Monte-Carlo gradient of ELBO, contributed immensely to the success
- Note: Most of these ideas apply also to Variational EM
- Many VI and advanced VI algorithms are implemented in probabilistic programming packages (e.g., Stan, Tensorflow Probability, etc), making VI a painless exercise even for complex models

Variational Inference: Some Comments

- Many probabilistic models (deep/non-deep) nowadays rely on VI to do tractable inference
- Even mean-field for locally-conjugate models has many applications in lots of probabilistic models
  - This + SVI gives excellent scalability
- Stoch. opt., auto. diff., Monte-Carlo gradient of ELBO, contributed immensely to the success
- Note: Most of these ideas apply also to Variational EM
- Many VI and advanced VI algorithms are implemented in probabilistic programming packages (e.g., Stan, Tensorflow Probability, etc), making VI a painless exercise even for complex models
- Still a very active area of research, especially for doing VI in complex models
  - Models with discrete latent variables
  - Reducing the variance in Monte-Carlo estimate of ELBO gradients