Approximate Inference: Variational Bayes Inference (2)

Piyush Rai

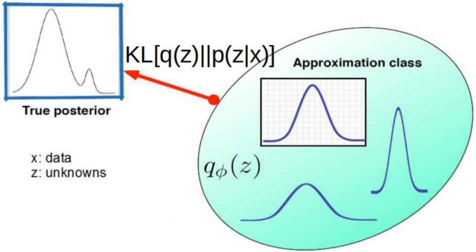
Probabilistic Machine Learning (CS772A)

Oct 12, 2017

Recap

Variational Bayes (VB) or Variational Inference (VI)

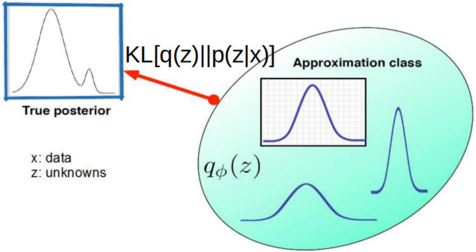
- Assume an approximation class of distributions $\{q(\boldsymbol{z}|\phi)\}$ parameterized by free parameters ϕ
- Approximate the true distribution $p(\boldsymbol{z}|\boldsymbol{x})$ by finding the "closest" $q(\boldsymbol{z}|\phi)$ from this class



Picture courtesy: Modified from the original pic by Shakir Mohamed

Variational Bayes (VB) or Variational Inference (VI)

- Assume an approximation class of distributions $\{q(\mathbf{z}|\phi)\}$ parameterized by free parameters ϕ
- Approximate the true distribution $p(\mathbf{z}|\mathbf{x})$ by finding the "closest" $q(\mathbf{z}|\phi)$ from this class

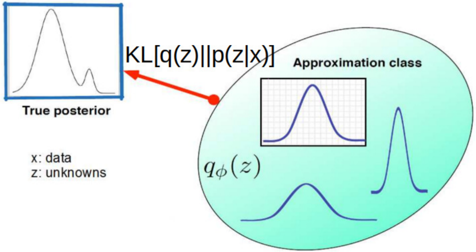


- ϕ (variational parameters) is like a "knob" that we have to tune to find the best matching $q(\mathbf{z}|\phi)$

Picture courtesy: Modified from the original pic by Shakir Mohamed

Variational Bayes (VB) or Variational Inference (VI)

- Assume an approximation class of distributions $\{q(\boldsymbol{z}|\phi)\}$ parameterized by free parameters ϕ
- Approximate the true distribution $p(\boldsymbol{z}|\boldsymbol{x})$ by finding the "closest" $q(\boldsymbol{z}|\phi)$ from this class

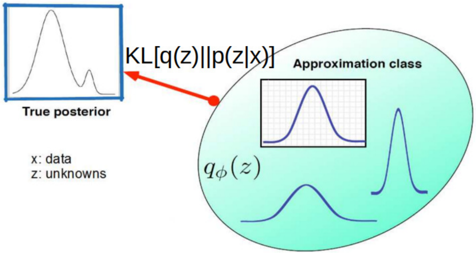


- ϕ (variational parameters) is like a "knob" that we have to tune to find the best matching $q(\boldsymbol{z}|\phi)$
- $\phi^* = \arg\min_\phi \text{KL}[q_\phi(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})]$

Picture courtesy: Modified from the original pic by Shakir Mohamed

Variational Bayes (VB) or Variational Inference (VI)

- Assume an approximation class of distributions $\{q(\boldsymbol{z}|\phi)\}$ parameterized by free parameters ϕ
- Approximate the true distribution $p(\boldsymbol{z}|\boldsymbol{x})$ by finding the "closest" $q(\boldsymbol{z}|\phi)$ from this class

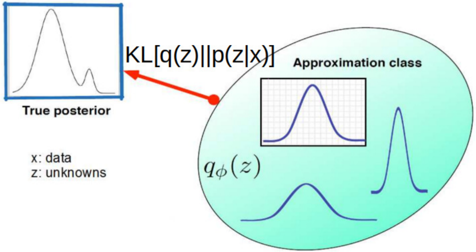


- ϕ (variational parameters) is like a "knob" that we have to tune to find the best matching $q(\boldsymbol{z}|\phi)$
- $\phi^* = \arg\min_\phi \text{KL}[q_\phi(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})]$: Approximate inference now becomes an optimization problem!

Picture courtesy: Modified from the original pic by Shakir Mohamed

Variational Bayes (VB) or Variational Inference (VI)

- Assume an approximation class of distributions $\{q(\boldsymbol{z}|\phi)\}$ parameterized by free parameters ϕ
- Approximate the true distribution $p(\boldsymbol{z}|\boldsymbol{x})$ by finding the "closest" $q(\boldsymbol{z}|\phi)$ from this class



- ϕ (variational parameters) is like a "knob" that we have to tune to find the best matching $q(\boldsymbol{z}|\phi)$
- $\phi^* = \arg\min_\phi \mathrm{KL}[q_\phi(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{x})]$: Approximate inference now becomes an optimization problem!
- Even though we don't know $p(\boldsymbol{z}|\boldsymbol{x})$, we can solve the above problem by maximizing the **ELBO**

Picture courtesy: Modified from the original pic by Shakir Mohamed

Variational Bayes (VB) Inference

- The key identity central to VB inference is the following (holds for any choice of q)

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \mathrm{KL}(q\|p)$$

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

$$\mathrm{KL}(q\|p) = -\int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

Variational Bayes (VB) Inference

- The key identity central to VB inference is the following (holds for any choice of q)

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \mathrm{KL}(q\|p)$$

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

$$\mathrm{KL}(q\|p) = -\int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

- **An Observation:** The L.H.S. $\log p(\mathbf{X})$ is a constant w.r.t. $\mathbf{Z}$

Variational Bayes (VB) Inference

- The key identity central to VB inference is the following (holds for any choice of q)

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \mathrm{KL}(q\|p)$$

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

$$\mathrm{KL}(q\|p) = -\int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

- **An Observation:** The L.H.S. $\log p(\mathbf{X})$ is a constant w.r.t. $\mathbf{Z}$
- Thus minimizing $KL(q||p)$ = maximizing $\mathcal{L}(q)$

Variational Bayes (VB) Inference

- The key identity central to VB inference is the following (holds for any choice of q)

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \mathrm{KL}(q\|p)$$

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

$$\mathrm{KL}(q\|p) = -\int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

- **An Observation:** The L.H.S. $\log p(\mathbf{X})$ is a constant w.r.t. $\mathbf{Z}$
- Thus minimizing $KL(q||p)$ = maximizing $\mathcal{L}(q)$
- Note: Since $KL \geq 0$, $\mathcal{L}(q)$ is a lower bound on the log-evidence $\log p(\mathbf{X})$ of the model m

Variational Bayes (VB) Inference

- The key identity central to VB inference is the following (holds for any choice of q)

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \mathrm{KL}(q\|p)$$

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

$$\mathrm{KL}(q\|p) = -\int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

- **An Observation:** The L.H.S. $\log p(\mathbf{X})$ is a constant w.r.t. $\mathbf{Z}$
- Thus minimizing $KL(q||p)$ = maximizing $\mathcal{L}(q)$
- Note: Since $KL \geq 0$, $\mathcal{L}(q)$ is a lower bound on the log-evidence $\log p(\mathbf{X})$ of the model m

$$\log p(\mathbf{X}|m) \geq \mathcal{L}(q)$$

Variational Bayes (VB) Inference

- The key identity central to VB inference is the following (holds for any choice of q)

$$\ln p(\mathbf{X}) = \mathcal{L}(q) + \mathrm{KL}(q||p)$$

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X},\mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

$$\mathrm{KL}(q||p) = -\int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z}|\mathbf{X})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

- **An Observation:** The L.H.S. $\log p(\mathbf{X})$ is a constant w.r.t. $\mathbf{Z}$
- Thus minimizing $KL(q||p)$ = maximizing $\mathcal{L}(q)$
- Note: Since $KL \geq 0$, $\mathcal{L}(q)$ is a lower bound on the log-evidence $\log p(\mathbf{X})$ of the model m

$$\log p(\mathbf{X}|m) \geq \mathcal{L}(q)$$

- Therefore $\mathcal{L}(q)$ is also known as the **Evidence Lower Bound** **(ELBO)**

VB by Maximizing the ELBO

- VB finds the q distribution that maximizes the ELBO

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

VB by Maximizing the ELBO

- VB finds the q distribution that maximizes the ELBO

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

- Note that q depends on the variational parameters ϕ. Expanding, we get

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

VB by Maximizing the ELBO

- VB finds the q distribution that maximizes the ELBO

$$\mathcal{L}(q) \quad = \quad \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

- Note that q depends on the variational parameters ϕ. Expanding, we get

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

- Maximizing $\mathcal{L}(q)$ w.r.t. q (equivalently ϕ) can still be hard (note the expectation w.r.t. q)

VB by Maximizing the ELBO

- VB finds the q distribution that maximizes the ELBO

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

- Note that q depends on the variational parameters ϕ. Expanding, we get

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

- Maximizing $\mathcal{L}(q)$ w.r.t. q (equivalently ϕ) can still be hard (note the expectation w.r.t. q)
- Some of the ways to make this problem easier

VB by Maximizing the ELBO

- VB finds the q distribution that maximizes the ELBO

$$\mathcal{L}(q) \quad = \quad \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X},\mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

- Note that q depends on the variational parameters ϕ. Expanding, we get

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

- Maximizing $\mathcal{L}(q)$ w.r.t. q (equivalently ϕ) can still be hard (note the expectation w.r.t. q)
- Some of the ways to make this problem easier
  1. Restricting the form of the q distribution, e.g., mean-field VB inference

VB by Maximizing the ELBO

- VB finds the q distribution that maximizes the ELBO

$$\mathcal{L}(q) = \int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{X}, \mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

- Note that q depends on the variational parameters ϕ. Expanding, we get

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

- Maximizing $\mathcal{L}(q)$ w.r.t. q (equivalently ϕ) can still be hard (note the expectation w.r.t. q)
- Some of the ways to make this problem easier
  1. Restricting the form of the q distribution, e.g., mean-field VB inference
  2. Using Monte-Carlo approximation of the expectation/gradient of the ELBO

Mean-Field VB Inference

- Suppose we partition the latent variables $\mathbf{Z}$ into M groups $\mathbf{Z}_1, \ldots, \mathbf{Z}_M$

Mean-Field VB Inference

- Suppose we partition the latent variables $\mathbf{Z}$ into M groups $\mathbf{Z}_1, \ldots, \mathbf{Z}_M$
- Assume our approximation $q(\mathbf{Z})$ factorizes over these groups

$$q(\mathbf{Z}|\phi) = \prod_{i=1}^{M} q(\mathbf{Z}_i|\phi_i)$$

Mean-Field VB Inference

- Suppose we partition the latent variables $\mathbf{Z}$ into M groups $\mathbf{Z}_1, \ldots, \mathbf{Z}_M$
- Assume our approximation $q(\mathbf{Z})$ factorizes over these groups

$$q(\mathbf{Z}|\phi) = \prod_{i=1}^{M} q(\mathbf{Z}_i|\phi_i)$$

- In mean-field VB, learning the optimal q reduces to learning the optimal $q_1, \ldots, q_M$

Mean-Field VB Inference

- Suppose we partition the latent variables $\mathbf{Z}$ into M groups $\mathbf{Z}_1, \ldots, \mathbf{Z}_M$
- Assume our approximation $q(\mathbf{Z})$ factorizes over these groups

$$q(\mathbf{Z}|\phi) = \prod_{i=1}^{M} q(\mathbf{Z}_i|\phi_i)$$

- In mean-field VB, learning the optimal q reduces to learning the optimal $q_1, \ldots, q_M$
- Can be done via one of the following two (equivalent) ways

Mean-Field VB Inference

- Suppose we partition the latent variables $\mathbf{Z}$ into M groups $\mathbf{Z}_1, \ldots, \mathbf{Z}_M$
- Assume our approximation $q(\mathbf{Z})$ factorizes over these groups

$$q(\mathbf{Z}|\phi) = \prod_{i=1}^{M} q(\mathbf{Z}_i|\phi_i)$$

- In mean-field VB, learning the optimal q reduces to learning the optimal $q_1, \ldots, q_M$
- Can be done via one of the following two (equivalent) ways
  - Compute each optimal q_j using the following expression (saw the derivation last time)

Mean-Field VB Inference

- Suppose we partition the latent variables $\mathbf{Z}$ into M groups $\mathbf{Z}_1, \ldots, \mathbf{Z}_M$
- Assume our approximation $q(\mathbf{Z})$ factorizes over these groups

$$q(\mathbf{Z}|\phi) = \prod_{i=1}^{M} q(\mathbf{Z}_i|\phi_i)$$

- In mean-field VB, learning the optimal q reduces to learning the optimal $q_1, \ldots, q_M$
- Can be done via one of the following two (equivalent) ways
  - Compute each optimal q_j using the following expression (saw the derivation last time)

$$q_j^\star(\mathbf{Z}_j) = \frac{\exp\left(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X}, \mathbf{Z})]\right)}{\int \exp\left(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X}, \mathbf{Z})]\right)\, \mathrm{d}\mathbf{Z}_j}$$

Mean-Field VB Inference

- Suppose we partition the latent variables $\mathbf{Z}$ into M groups $\mathbf{Z}_1, \ldots, \mathbf{Z}_M$
- Assume our approximation $q(\mathbf{Z})$ factorizes over these groups

$$q(\mathbf{Z}|\phi) = \prod_{i=1}^{M} q(\mathbf{Z}_i|\phi_i)$$

- In mean-field VB, learning the optimal q reduces to learning the optimal $q_1, \ldots, q_M$
- Can be done via one of the following two (equivalent) ways
  - Compute each optimal q_j using the following expression (saw the derivation last time)

$$q_j^\star(\mathbf{Z}_j) = \frac{\exp\left(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X}, \mathbf{Z})]\right)}{\int \exp\left(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X}, \mathbf{Z})]\right)\, \mathrm{d}\mathbf{Z}_j}$$

  - Assume some parametric form of each q_i (ϕ_i's will be params) and write down the ELBO

$$\mathcal{L}(q) = \mathcal{L}(\phi_1, \ldots, \phi_M) \quad = \quad \int q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z}) d\mathbf{Z} - \int q(\mathbf{Z}) \log q(\mathbf{Z}) d\mathbf{Z}$$

Mean-Field VB Inference

- Suppose we partition the latent variables $\mathbf{Z}$ into M groups $\mathbf{Z}_1, \ldots, \mathbf{Z}_M$
- Assume our approximation $q(\mathbf{Z})$ factorizes over these groups

$$q(\mathbf{Z}|\phi) = \prod_{i=1}^{M} q(\mathbf{Z}_i|\phi_i)$$

- In mean-field VB, learning the optimal q reduces to learning the optimal $q_1, \ldots, q_M$
- Can be done via one of the following two (equivalent) ways
  - Compute each optimal q_j using the following expression (saw the derivation last time)

$$q_j^\star(\mathbf{Z}_j) = \frac{\exp\left(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})]\right)}{\int \exp\left(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X},\mathbf{Z})]\right)\,\mathrm{d}\mathbf{Z}_j}$$

  - Assume some parametric form of each q_i (ϕ_i's will be params) and write down the ELBO

$$\mathcal{L}(q) = \mathcal{L}(\phi_1, \ldots, \phi_M) \quad = \quad \int q(\mathbf{Z}) \log p(\mathbf{X},\mathbf{Z}) d\mathbf{Z} - \int q(\mathbf{Z}) \log q(\mathbf{Z}) d\mathbf{Z}$$

  .. and compute the optimal values for variational parameters $\phi_1, \ldots, \phi_M$ by taking derivatives..

Mean-Field VB Inference

- Suppose we partition the latent variables $\mathbf{Z}$ into M groups $\mathbf{Z}_1, \ldots, \mathbf{Z}_M$
- Assume our approximation $q(\mathbf{Z})$ factorizes over these groups

$$q(\mathbf{Z}|\phi) = \prod_{i=1}^{M} q(\mathbf{Z}_i|\phi_i)$$

- In mean-field VB, learning the optimal q reduces to learning the optimal $q_1, \ldots, q_M$
- Can be done via one of the following two (equivalent) ways
  - Compute each optimal q_j using the following expression (saw the derivation last time)

$$q_j^\star(\mathbf{Z}_j) = \frac{\exp\left(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X}, \mathbf{Z})]\right)}{\int \exp\left(\mathbb{E}_{i\neq j}[\ln p(\mathbf{X}, \mathbf{Z})]\right)\, \mathrm{d}\mathbf{Z}_j}$$

  - Assume some parametric form of each q_i (ϕ_i's will be params) and write down the ELBO

$$\mathcal{L}(q) = \mathcal{L}(\phi_1, \ldots, \phi_M) \quad = \quad \int q(\mathbf{Z}) \log p(\mathbf{X}, \mathbf{Z}) d\mathbf{Z} - \int q(\mathbf{Z}) \log q(\mathbf{Z}) d\mathbf{Z}$$

    .. and compute the optimal values for variational parameters $\phi_1, \ldots, \phi_M$ by taking derivatives..
- For both cases, deriving mean-field VB updates is easy if the model has local conjugacy

Some Properties of VB

Recall that VB is equivalent to finding q by minimizing $\text{KL}(q||p)$

$$\mathrm{KL}(q\|p) = -\int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

If the true posterior p is very small in some region then, to minimize $\text{KL}(q||p)$, the approx. dist. q will also have to be very small (otherwise KL will be very large)

Some Properties of VB

Recall that VB is equivalent to finding q by minimizing $\text{KL}(q||p)$

$$\text{KL}(q||p) = -\int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

If the true posterior p is very small in some region then, to minimize $\text{KL}(q||p)$, the approx. dist. q will also have to be very small (otherwise KL will be very large)

This has two key consequences for VB

- Underestimates the variances of the true posterior
- For multimodal posteriors, VB locks onto one of the modes

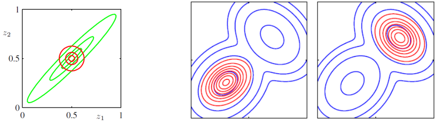


Figure: (Left) Zero-Forcing Property of VB, (Right) For multi-modal posterior, VB locks onto one of the models

Some Properties of VB

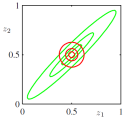
Recall that VB is equivalent to finding q by minimizing $\mathrm{KL}(q||p)$

$$\mathrm{KL}(q||p) = -\int q(\mathbf{Z}) \ln \left\{ \frac{p(\mathbf{Z})}{q(\mathbf{Z})} \right\} \mathrm{d}\mathbf{Z}$$

If the true posterior p is very small in some region then, to minimize $\mathrm{KL}(q||p)$, the approx. dist. q will also have to be very small (otherwise KL will be very large)

This has two key consequences for VB

- Underestimates the variances of the true posterior
- For multimodal posteriors, VB locks onto one of the modes



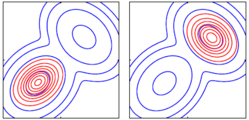


Figure: (Left) Zero-Forcing Property of VB, (Right) For multi-modal posterior, VB locks onto one of the models

Note: Some other inference methods, e.g., Expectation Propagation (EP) can avoid this behavior

Today's Plan

Some recent advances in VB inference

Today's Plan

Some recent advances in VB inference

- Non-conjugate models. Lots of work on this. Will look at an approach called "Black-box" VI
  - Based on the idea of Monte-Carlo approximation of the ELBO gradient

Today's Plan

Some recent advances in VB inference

- Non-conjugate models. Lots of work on this. Will look at an approach called "Black-box" VI
  - Based on the idea of Monte-Carlo approximation of the ELBO gradient
- Scaling up VB for large datasets (Stochastic/Online VB)
  - Based on the idea of stochastic optimization techniques

Variational Inference for Non-Conjugate Models

Mean-Field VB for Non-Conjugate Models

- Several ways to handle this

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

Mean-Field VB for Non-Conjugate Models

- Several ways to handle this
  - Use conjugate approximations for the non-conjugate parts of the model and then do MF-VB

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

Mean-Field VB for Non-Conjugate Models

- Several ways to handle this
  - Use conjugate approximations for the non-conjugate parts of the model and then do MF-VB
  - **Black-box Variational Inference:** Approximate ELBO derivatives using Monte-Carlo methods

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

Mean-Field VB for Non-Conjugate Models

- Several ways to handle this
  - Use conjugate approximations for the non-conjugate parts of the model and then do MF-VB
  - **Black-box Variational Inference:** Approximate ELBO derivatives using Monte-Carlo methods
- Black-box Variational Inference (BBVI) 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)

Mean-Field VB for Non-Conjugate Models

- Several ways to handle this
  - Use conjugate approximations for the non-conjugate parts of the model and then do MF-VB
  - **Black-box Variational Inference:** Approximate ELBO derivatives using Monte-Carlo methods
- Black-box Variational Inference (BBVI) 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)

Mean-Field VB for Non-Conjugate Models

- Several ways to handle this
  - Use conjugate approximations for the non-conjugate parts of the model and then do MF-VB
  - **Black-box Variational Inference:** Approximate ELBO derivatives using Monte-Carlo methods
- Black-box Variational Inference (BBVI) 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 in terms of expectation of gradient of $\log q(\mathbf{Z}|\phi)$

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

Mean-Field VB for Non-Conjugate Models

- Several ways to handle this
  - Use conjugate approximations for the non-conjugate parts of the model and then do MF-VB
  - **Black-box Variational Inference:** Approximate ELBO derivatives using Monte-Carlo methods
- Black-box Variational Inference (BBVI) 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 in terms of expectation of gradient of $\log q(\mathbf{Z}|\phi)$
- 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)

Proof of BBVI Identity

- The ELBO gradient can be written as

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

Proof of BBVI Identity

- The ELBO gradient can be written as

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

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

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

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

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

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

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

- 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

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

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
- 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
- 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
- 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
- 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

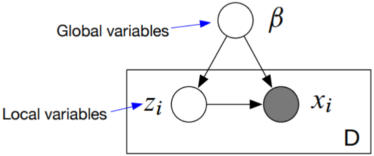
$$\nabla_\phi \mathcal{L}(q) \approx \frac{1}{S}\sum_{s=1}^{S} \nabla_\phi \log q(\mathbf{Z}_s|\phi)(\log p(\mathbf{X}, \mathbf{Z}_s) - \log q(\mathbf{Z}_s|\phi))$$

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

Scalable (Online) Variational Inference

A Generic Probabilistic Model

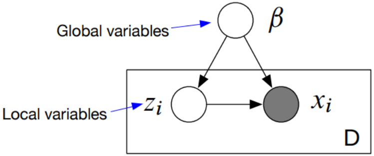
- Assume D data points $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D\}$ generated via a probabilistic model



- Assume local latent variables $\mathbf{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_D\}$, one per data point ($\boldsymbol{z}_i$ for $\boldsymbol{x}_i$, $i = 1, \ldots, D$)
- Assume global latent variables β shared by all data points
- Probability distribution of data point $\boldsymbol{x}_i$ only depends on $\boldsymbol{z}_i$ and β

A Generic Probabilistic Model

- Assume D data points $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D\}$ generated via a probabilistic model

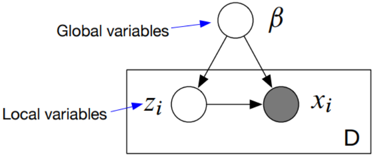


- Assume local latent variables $\mathbf{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_D\}$, one per data point ($\boldsymbol{z}_i$ for $\boldsymbol{x}_i$, $i = 1, \ldots, D$)
- Assume global latent variables $\boldsymbol{\beta}$ shared by all data points
- Probability distribution of data point $\boldsymbol{x}_i$ only depends on $\boldsymbol{z}_i$ and $\boldsymbol{\beta}$
- The joint distribution for this model admits the following factorization

$$p(\mathbf{X}, \mathbf{Z}, \beta) = p(\beta) \prod_{i=1}^{D} p(x_i, z_i | \beta)$$

A Generic Probabilistic Model

- Assume D data points $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D\}$ generated via a probabilistic model



- Assume local latent variables $\mathbf{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_D\}$, one per data point ($\boldsymbol{z}_i$ for $\boldsymbol{x}_i$, $i = 1, \ldots, D$)
- Assume global latent variables $\boldsymbol{\beta}$ shared by all data points
- Probability distribution of data point $\boldsymbol{x}_i$ only depends on $\boldsymbol{z}_i$ and $\boldsymbol{\beta}$
- The joint distribution for this model admits the following factorization

$$p(\mathbf{X}, \mathbf{Z}, \beta) = p(\beta) \prod_{i=1}^{D} p(x_i, z_i | \beta) = p(\beta) \prod_{i=1}^{D} p(x_i | z_i, \beta) p(z_i)$$

Mean-Field VB

- Our goal is to infer the posterior distribution $p(\mathbf{Z}, \beta|\mathbf{X})$. Intractable in general.
- Let's approximate $p(\mathbf{Z}, \beta|\mathbf{X})$ using a mean-field distribution $q(\mathbf{Z}, \beta)$

$$q(\mathbf{Z}, \beta) = q(\beta)\prod_{i=1}^{D} q(\boldsymbol{z}_i)$$

Mean-Field VB

- Our goal is to infer the posterior distribution $p(\mathbf{Z}, \boldsymbol{\beta}|\mathbf{X})$. Intractable in general.
- Let's approximate $p(\mathbf{Z}, \beta|\mathbf{X})$ using a mean-field distribution $q(\mathbf{Z}, \beta)$

$$q(\mathbf{Z}, \beta) = q(\beta) \prod_{i=1}^{D} q(\boldsymbol{z}_i)$$

- The log joint probability has a simple form as a summation over all data points

$$\ln p(\boldsymbol{x}, \boldsymbol{z}, \beta) = \sum_{i=1}^{D} \ln p(x_i, z_i|\beta) \; + \; \ln p(\beta)$$

Mean-Field VB

- Our goal is to infer the posterior distribution $p(\mathbf{Z}, \beta|\mathbf{X})$. Intractable in general.
- Let's approximate $p(\mathbf{Z}, \beta|\mathbf{X})$ using a mean-field distribution $q(\mathbf{Z}, \beta)$

$$q(\mathbf{Z}, \beta) = q(\beta) \prod_{i=1}^{D} q(\boldsymbol{z}_i)$$

- The log joint probability has a simple form as a summation over all data points

$$\ln p(\boldsymbol{x}, \boldsymbol{z}, \beta) = \sum_{i=1}^{D} \ln p(x_i, z_i|\beta) \; + \; \ln p(\beta)$$

- The ELBO $\mathcal{L} = \mathbb{E}_q \left[\ln \frac{p(\mathbf{X},\mathbf{Z},\beta)}{q(\mathbf{Z},\beta)} \right]$ for the model can be written as

$$\mathcal{L} = \sum_{i=1}^{D} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

Mean-Field VB

- Our goal is to infer the posterior distribution $p(\mathbf{Z}, \beta|\mathbf{X})$. Intractable in general.
- Let's approximate $p(\mathbf{Z}, \beta|\mathbf{X})$ using a mean-field distribution $q(\mathbf{Z}, \beta)$

$$q(\mathbf{Z}, \beta) = q(\beta) \prod_{i=1}^{D} q(\boldsymbol{z}_i)$$

- The log joint probability has a simple form as a summation over all data points

$$\ln p(\boldsymbol{x}, \boldsymbol{z}, \beta) = \sum_{i=1}^{D} \ln p(x_i, z_i|\beta) \; + \; \ln p(\beta)$$

- The ELBO $\mathcal{L} = \mathbb{E}_q \left[\ln \frac{p(\mathbf{X},\mathbf{Z},\beta)}{q(\mathbf{Z},\beta)}\right]$ for the model can be written as

$$\mathcal{L} = \sum_{i=1}^{D} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)}\right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)}\right]$$

- Can now do mean-field VB by optimizing w.r.t. $q(\boldsymbol{z}_i)$, $\forall i$, and $q(\beta)$, until convergence

Mean-Field VB

- The basic mean-field VB is a batch algorithm (each iteration operates on all the data)
- Each iteration has to look at every data point $\boldsymbol{x}_i$ and infer the corresponding $q(\boldsymbol{z}_i)$, $\forall i$

Batch variational inference

1. For $i = 1, \ldots, D$, optimize $q(z_i)$
2. Optimize $q(\beta)$ ← Depends on all the $q(z_i)$
3. Repeat

Mean-Field VB

- The basic mean-field VB is a batch algorithm (each iteration operates on all the data)
- Each iteration has to look at every data point $\boldsymbol{x}_i$ and infer the corresponding $q(\boldsymbol{z}_i)$, $\forall i$

<u>Batch variational inference</u>

1. For $i = 1, \ldots, D$, optimize $q(z_i)$
2. Optimize $q(\beta)$ ← Depends on all the $q(z_i)$
3. Repeat

- This can be very slow is the number of data points D is very large
  - Before updating the global variables β, we must update all the local variables $\boldsymbol{z}_i$'s

Mean-Field VB

- The basic mean-field VB is a batch algorithm (each iteration operates on all the data)
- Each iteration has to look at every data point $\boldsymbol{x}_i$ and infer the corresponding $q(\boldsymbol{z}_i)$, $\forall i$

Batch variational inference

1. For $i = 1, \ldots, D$, optimize $q(z_i)$
2. Optimize $q(\beta)$ ← Depends on all the $q(z_i)$
3. Repeat

- This can be very slow is the number of data points D is very large
  - Before updating the global variables β, we must update all the local variables $\boldsymbol{z}_i$'s
- Would like to have a faster inference algorithm that scales nicely with number of data points

Stochastic Variational Inference (SVI)

- Based on the well-known idea of stochastic optimization

Stochastic Variational Inference - Hoffman et al (2013)

Stochastic Variational Inference (SVI)

- Based on the well-known idea of stochastic optimization
- In each VB iteration, let's use only a small mini-batch of data points (chosen randomly)

Stochastic Variational Inference - Hoffman et al (2013)

Stochastic Variational Inference (SVI)

- Based on the well-known idea of stochastic optimization
- In each VB iteration, let's use only a small mini-batch of data points (chosen randomly)

Stochastic variational inference

1. Randomly select a subset of local data, $S_t \subset \{1, \ldots, D\}$
2. Construct the scaled variational objective function

$$\mathcal{L}_t = \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i | \beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

3. Optimize each $q(z_i)$ in $\mathcal{L}_t$ *only*
4. Update the parameters of $q(\beta|\psi)$ using a gradient step

$$q(\beta|\psi) \quad \rightarrow \quad \psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$$

5. Repeat

Stochastic Variational Inference - Hoffman et al (2013)

Stochastic Variational Inference (SVI)

- Based on the well-known idea of stochastic optimization
- In each VB iteration, let's use only a small mini-batch of data points (chosen randomly)

Stochastic variational inference

1. Randomly select a subset of local data, $S_t \subset \{1, \ldots, D\}$
2. Construct the scaled variational objective function

$$\mathcal{L}_t = \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

3. Optimize each $q(z_i)$ in $\mathcal{L}_t$ *only*
4. Update the parameters of $q(\beta|\psi)$ using a gradient step

$$q(\beta|\psi) \quad \rightarrow \quad \psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$$

5. Repeat

- Similar in spirit to online EM

Stochastic Variational Inference - Hoffman et al (2013)

Stochastic Variational Inference (SVI)

- Based on the well-known idea of stochastic optimization
- In each VB iteration, let's use only a small mini-batch of data points (chosen randomly)

Stochastic variational inference

1. Randomly select a subset of local data, $S_t \subset \{1, \ldots, D\}$
2. Construct the scaled variational objective function

$$\mathcal{L}_t = \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

3. Optimize each $q(z_i)$ in $\mathcal{L}_t$ *only*
4. Update the parameters of $q(\beta|\psi)$ using a gradient step

$$q(\beta|\psi) \quad \rightarrow \quad \psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$$

5. Repeat

- Similar in spirit to online EM
- ρ_t is a learning rate. M_t (a "pre-conditioner") will be defined later

Stochastic Variational Inference - Hoffman et al (2013)

Stochastic Variational Inference (SVI)

- Based on the well-known idea of stochastic optimization
- In each VB iteration, let's use only a small mini-batch of data points (chosen randomly)

Stochastic variational inference

1. Randomly select a subset of local data, $S_t \subset \{1, \ldots, D\}$
2. Construct the scaled variational objective function

$$\mathcal{L}_t = \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

3. Optimize each $q(z_i)$ in $\mathcal{L}_t$ *only*
4. Update the parameters of $q(\beta|\psi)$ using a gradient step

$$q(\beta|\psi) \quad \rightarrow \quad \psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$$

5. Repeat

- Similar in spirit to online EM
- ρ_t is a learning rate. M_t (a "pre-conditioner") will be defined later
- Note: In step-4, instead of solving for $q(\beta|\psi)$ analytically, we are using a gradient method

Stochastic Variational Inference - Hoffman et al (2013)

ELBO on the Mini-Batch

- The ELBO on the full data

$$\mathcal{L} = \sum_{i=1}^{D} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

ELBO on the Mini-Batch

- The ELBO on the full data

$$\mathcal{L} = \sum_{i=1}^{D} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

- The (scaled) ELBO on the random subset (mini-batch) of the data chosen in iteration t

$$\mathcal{L}_t = \frac{D}{|S_t|} \sum_{i=1}^{D} \mathbb{1}(d \in S_t) \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

ELBO on the Mini-Batch

- The ELBO on the full data

$$\mathcal{L} = \sum_{i=1}^{D} \mathbb{E}_q\left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)}\right] + \mathbb{E}_q\left[\ln \frac{p(\beta)}{q(\beta)}\right]$$

- The (scaled) ELBO on the random subset (mini-batch) of the data chosen in iteration t

$$\mathcal{L}_t = \frac{D}{|S_t|}\sum_{i=1}^{D} \mathbb{1}(d \in S_t)\mathbb{E}_q\left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)}\right] + \mathbb{E}_q\left[\ln \frac{p(\beta)}{q(\beta)}\right]$$

- The expectation of the mini-batch ELBO (expectation w.r.t. the random selection of mini-batch)

$$\mathbb{E}_P[\mathcal{L}_t] = \frac{D}{|S_t|}\sum_{i=1}^{D} \underbrace{\mathbb{E}_P[\mathbb{1}(d \in S_t)]}_{P(d\in S_t)}\mathbb{E}_q\left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)}\right] + \mathbb{E}_q\left[\ln \frac{p(\beta)}{q(\beta)}\right]$$

ELBO on the Mini-Batch

- The ELBO on the full data

$$\mathcal{L} = \sum_{i=1}^{D} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

- The (scaled) ELBO on the random subset (mini-batch) of the data chosen in iteration t

$$\mathcal{L}_t = \frac{D}{|S_t|} \sum_{i=1}^{D} \mathbb{1}(d \in S_t) \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

- The expectation of the mini-batch ELBO (expectation w.r.t. the random selection of mini-batch)

$$\mathbb{E}_P[\mathcal{L}_t] = \frac{D}{|S_t|} \sum_{i=1}^{D} \underbrace{\mathbb{E}_P[\mathbb{1}(d \in S_t)]}_{P(d \in S_t)} \mathbb{E}_q \left[\ln \frac{p(x_i, z_i|\beta)}{q(z_i)} \right] + \mathbb{E}_q \left[\ln \frac{p(\beta)}{q(\beta)} \right]$$

- Note that $P(d \in S_t) = \frac{|S_t|}{D}$. Therefore $\mathbb{E}_P[\mathcal{L}_t] = \mathcal{L}$ (this is good news!)

Stochastic Updates for SVI

- For our stochastic update of ψ for updating $q(\beta|\psi)$, i.e., $\psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$

$$\mathbb{E}_P[\psi_t] = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathbb{E}_P[\mathcal{L}_t] (= \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L})$$

Stochastic Updates for SVI

- For our stochastic update of ψ for updating $q(\beta|\psi)$, i.e., $\psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$

$$\mathbb{E}_P[\psi_t] = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathbb{E}_P[\mathcal{L}_t] (= \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L})$$

- Therefore the stochastic gradient computed using S_t is unbiased

Stochastic Updates for SVI

- For our stochastic update of ψ for updating $q(\beta|\psi)$, i.e., $\psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$

$$\mathbb{E}_P[\psi_t] = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathbb{E}_P[\mathcal{L}_t] (= \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L})$$

- Therefore the stochastic gradient computed using S_t is unbiased
- We now basically have an stochastic gradient method to update ψ

Stochastic Updates for SVI

- For our stochastic update of ψ for updating $q(\beta|\psi)$, i.e., $\psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$

$$\mathbb{E}_P[\psi_t] = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathbb{E}_P[\mathcal{L}_t] (= \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L})$$

- Therefore the stochastic gradient computed using S_t is unbiased
- We now basically have an stochastic gradient method to update ψ
- Note: The learning rate ρ_t must satisfy $\sum_{t=1}^{\infty} \rho_t = \infty$ and $\sum_{t=1}^{\infty} \rho_t^2 < \infty$

Stochastic Updates for SVI

- For our stochastic update of ψ for updating $q(\beta|\psi)$, i.e., $\psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$

$$\mathbb{E}_P[\psi_t] = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathbb{E}_P[\mathcal{L}_t] (= \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L})$$

- Therefore the stochastic gradient computed using S_t is unbiased
- We now basically have an stochastic gradient method to update ψ
- Note: The learning rate ρ_t must satisfy $\sum_{t=1}^{\infty} \rho_t = \infty$ and $\sum_{t=1}^{\infty} \rho_t^2 < \infty$
  - Setting $\rho_t = \frac{1}{(t_0+t)^\kappa}$ with $\kappa \in (0.5, 1)$ ensures this. t_0 is some positive number.

Stochastic Updates for SVI

- For our stochastic update of ψ for updating $q(\beta|\psi)$, i.e., $\psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$

$$\mathbb{E}_P[\psi_t] = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathbb{E}_P[\mathcal{L}_t] (= \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L})$$

- Therefore the stochastic gradient computed using S_t is unbiased
- We now basically have an stochastic gradient method to update ψ
- Note: The learning rate ρ_t must satisfy $\sum_{t=1}^{\infty} \rho_t = \infty$ and $\sum_{t=1}^{\infty} \rho_t^2 < \infty$
  - Setting $\rho_t = \frac{1}{(t_0+t)^\kappa}$ with $\kappa \in (0.5, 1)$ ensures this. t_0 is some positive number.
- Note that SVI also gives us all the "local" $q(\boldsymbol{z}_i)$ distributions in the normal way. It's only the $q(\beta)$ distribution (which depends on the local distributions) that is inferred in an online fashion

SVI for Exponential Family Distributions

- Assuming the joint distribution of data $\boldsymbol{x}_i$ and local latent var. $\boldsymbol{z}_i$ to be exponential family dist.

$$p(\mathbf{X}, \mathbf{Z}|\beta) = \prod_{i=1}^{D} p(x_i, z_i|\beta) = \left[\prod_{i=1}^{D} h(x_i, z_i)\right] e^{\beta^T \sum_{i=1}^{D} t(x_i, z_i) - DA(\beta)}$$

SVI for Exponential Family Distributions

- Assuming the joint distribution of data $\boldsymbol{x}_i$ and local latent var. $\boldsymbol{z}_i$ to be exponential family dist.

$$p(\mathbf{X}, \mathbf{Z}|\beta) = \prod_{i=1}^{D} p(x_i, z_i|\beta) = \left[\prod_{i=1}^{D} h(x_i, z_i)\right] \mathrm{e}^{\beta^T \sum_{i=1}^{D} t(x_i, z_i) - DA(\beta)}$$

- Let's assume a conjugate prior on the global variables β

$$p(\beta) = f(\xi, \nu) e^{\beta^\top \xi - \nu A(\beta)}$$

SVI for Exponential Family Distributions

- Assuming the joint distribution of data $\boldsymbol{x}_i$ and local latent var. $\boldsymbol{z}_i$ to be exponential family dist.

$$p(\mathbf{X}, \mathbf{Z}|\beta) = \prod_{i=1}^{D} p(x_i, z_i|\beta) = \left[\prod_{i=1}^{D} h(x_i, z_i)\right] \mathrm{e}^{\beta^T \sum_{i=1}^{D} t(x_i, z_i) - DA(\beta)}$$

- Let's assume a conjugate prior on the global variables β

$$p(\beta) = f(\xi, \nu) e^{\beta^\top \xi - \nu A(\beta)}$$

- Let's assume our variational distribution $q(\beta)$ to have the same form as the prior $p(\beta)$

$$q(\beta) = f(\xi', \nu') e^{\beta^\top \xi' - \nu' A(\beta)}$$

SVI for Exponential Family Distributions

- Assuming the joint distribution of data $\boldsymbol{x}_i$ and local latent var. $\boldsymbol{z}_i$ to be exponential family dist.

$$p(\mathbf{X}, \mathbf{Z}|\beta) = \prod_{i=1}^{D} p(x_i, z_i|\beta) = \left[\prod_{i=1}^{D} h(x_i, z_i)\right] \mathrm{e}^{\beta^T \sum_{i=1}^{D} t(x_i, z_i) - DA(\beta)}$$

- Let's assume a conjugate prior on the global variables β

$$p(\beta) = f(\xi, \nu) e^{\beta^\top \xi - \nu A(\beta)}$$

- Let's assume our variational distribution $q(\beta)$ to have the same form as the prior $p(\beta)$

$$q(\beta) = f(\xi', \nu') e^{\beta^\top \xi' - \nu' A(\beta)}$$

- SVI will basically do stochastic optimization to learn the parameters ξ', ν' of variational distribution

SVI for Exponential Family Distributions

- Assuming the joint distribution of data $\boldsymbol{x}_i$ and local latent var. $\boldsymbol{z}_i$ to be exponential family dist.

$$p(\mathbf{X},\mathbf{Z}|\beta) = \prod_{i=1}^{D} p(x_i, z_i|\beta) = \left[\prod_{i=1}^{D} h(x_i, z_i)\right] \mathrm{e}^{\beta^T \sum_{i=1}^{D} t(x_i,z_i) - DA(\beta)}$$

- Let's assume a conjugate prior on the global variables β

$$p(\beta) = f(\xi,\nu)e^{\beta^\top \xi - \nu A(\beta)}$$

- Let's assume our variational distribution $q(\beta)$ to have the same form as the prior $p(\beta)$

$$q(\beta) = f(\xi',\nu')e^{\beta^\top \xi' - \nu' A(\beta)}$$

- SVI will basically do stochastic optimization to learn the parameters ξ', ν' of variational distribution
- We will now look at the general form of these updates for a generic model with exp. family dist.

SVI for Exponential Family Distributions

- If we were doing full batch updates for $q(\beta|\psi)$ (where $\psi = [\xi, \nu]$) using gradient methods then

$$\begin{bmatrix} \xi' \\ \nu' \end{bmatrix} \leftarrow \begin{bmatrix} \xi' \\ \nu' \end{bmatrix} + \rho_t M_t \nabla_{(\xi', \nu')} \mathcal{L}$$

SVI for Exponential Family Distributions

- If we were doing full batch updates for $q(\beta|\psi)$ (where $\psi = [\xi, \nu]$) using gradient methods then

$$\begin{bmatrix} \xi' \\ \nu' \end{bmatrix} \leftarrow \begin{bmatrix} \xi' \\ \nu' \end{bmatrix} + \rho_t M_t \nabla_{(\xi', \nu')} \mathcal{L}$$

- For this update, the part of $\mathcal{L}$ that depends on β is

$$\mathcal{L}_\beta = \sum_{i=1}^{D} \mathbb{E}_q[\ln p(x_i, z_i|\beta)] + \mathbb{E}_q[\ln p(\beta)] - \mathbb{E}_q[\ln q(\beta)]$$

SVI for Exponential Family Distributions

- If we were doing full batch updates for $q(\beta|\psi)$ (where $\psi = [\xi, \nu]$) using gradient methods then

$$\begin{bmatrix} \xi' \\ \nu' \end{bmatrix} \leftarrow \begin{bmatrix} \xi' \\ \nu' \end{bmatrix} + \rho_t M_t \nabla_{(\xi', \nu')} \mathcal{L}$$

- For this update, the part of $\mathcal{L}$ that depends on β is

$$\mathcal{L}_\beta = \sum_{i=1}^{D} \mathbb{E}_q[\ln p(x_i, z_i|\beta)] + \mathbb{E}_q[\ln p(\beta)] - \mathbb{E}_q[\ln q(\beta)]$$

- Plugging in the expressions for exponential family distributions (given on the previous slide)

$$\mathcal{L}_\beta = \mathbb{E}_q[\beta]^T \left(\sum_{i=1}^{D} \mathbb{E}[t(x_i, z_i)] + \xi - \xi' \right) - \mathbb{E}_q[A(\beta)](D + \nu - \nu') + \ln f(\xi', \nu') + \text{const.}$$

SVI for Exponential Family Distributions

- If we were doing full batch updates for $q(\beta|\psi)$ (where $\psi = [\xi, \nu]$) using gradient methods then

$$\begin{bmatrix} \xi' \\ \nu' \end{bmatrix} \leftarrow \begin{bmatrix} \xi' \\ \nu' \end{bmatrix} + \rho_t M_t \nabla_{(\xi', \nu')} \mathcal{L}$$

- For this update, the part of $\mathcal{L}$ that depends on β is

$$\mathcal{L}_\beta = \sum_{i=1}^{D} \mathbb{E}_q[\ln p(x_i, z_i|\beta)] + \mathbb{E}_q[\ln p(\beta)] - \mathbb{E}_q[\ln q(\beta)]$$

- Plugging in the expressions for exponential family distributions (given on the previous slide)

$$\mathcal{L}_\beta = \mathbb{E}_q[\beta]^T \left(\sum_{i=1}^{D} \mathbb{E}[t(x_i, z_i)] + \xi - \xi' \right) - \mathbb{E}_q[A(\beta)](D + \nu - \nu') + \ln f(\xi', \nu') + \text{const.}$$

- Note that it requires computing two expectations $\mathbb{E}_q[\beta]$ and $\mathbb{E}_q[A(\beta)]$

SVI for Exponential Family Distributions

- The expectations $\mathbb{E}_q[\beta]$ and $\mathbb{E}_q[A(\beta)]$ can be computed as follows

$$q(\beta) = f(\xi', \nu')\mathrm{e}^{\beta^T \xi' - \nu' A(\beta)}$$

$$\int \nabla_{\xi'} q(\beta) d\beta = 0, \qquad \int \frac{\partial}{\partial \nu'} q(\beta) d\beta = 0$$

$$\mathbb{E}_q[\beta] = -\nabla_{\xi'} \ln f(\xi', \nu'), \qquad \mathbb{E}_q[A(\beta)] = \frac{\partial \ln f(\xi', \nu')}{\partial \nu'}$$

- Exercise: Verify the above (using the fact that $q(\beta)$ is an exp-family dist.)

SVI for Exponential Family Distributions

- The expectations $\mathbb{E}_q[\beta]$ and $\mathbb{E}_q[A(\beta)]$ can be computed as follows

$$q(\beta) = f(\xi', \nu')\mathrm{e}^{\beta^T\xi' - \nu' A(\beta)}$$

$$\int \nabla_{\xi'} q(\beta) d\beta = 0, \qquad \int \frac{\partial}{\partial \nu'} q(\beta) d\beta = 0$$

$$\mathbb{E}_q[\beta] = -\nabla_{\xi'} \ln f(\xi', \nu'), \qquad \mathbb{E}_q[A(\beta)] = \frac{\partial \ln f(\xi', \nu')}{\partial \nu'}$$

- Exercise: Verify the above (using the fact that $q(\beta)$ is an exp-family dist.)
- These can then be plugged into the expression of $\mathcal{L}_\beta$

SVI for Exponential Family Distributions

- ELBO gradient (batch case)

$$\nabla_{(\xi',\nu')}\mathcal{L}_\beta = \begin{bmatrix} \nabla_{\xi'}\mathcal{L}_\beta \\ \frac{\partial}{\partial\nu'}\mathcal{L}_\beta \end{bmatrix} = -\begin{bmatrix} \nabla^2_{\xi'}\ln f(\xi',\nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'^2} \end{bmatrix} \begin{bmatrix} \sum_{i=1}^D \mathbb{E}_q[t(x_i,z_i)] + \xi - \xi' \\ D + \nu - \nu' \end{bmatrix}$$

SVI for Exponential Family Distributions

- ELBO gradient (batch case)

$$\nabla_{(\xi',\nu')}\mathcal{L}_\beta = \begin{bmatrix} \nabla_{\xi'}\mathcal{L}_\beta \\ \frac{\partial}{\partial\nu'}\mathcal{L}_\beta \end{bmatrix} = -\begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi',\nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'^2} \end{bmatrix} \begin{bmatrix} \sum_{i=1}^D \mathbb{E}_q[t(x_i,z_i)] + \xi - \xi' \\ D + \nu - \nu' \end{bmatrix}$$

- Since preconditioning matrix is p.s.d., setting gradient to zero gives closed-form batch VB update

$$\xi' = \xi + \sum_{i=1}^D \mathbb{E}[t(x_i,z_i)], \qquad \nu' = \nu + D$$

SVI for Exponential Family Distributions

- ELBO gradient (batch case)

$$\nabla_{(\xi',\nu')}\mathcal{L}_\beta = \begin{bmatrix} \nabla_{\xi'}\mathcal{L}_\beta \\ \frac{\partial}{\partial\nu'}\mathcal{L}_\beta \end{bmatrix} = -\begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi',\nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'^2} \end{bmatrix} \begin{bmatrix} \sum_{i=1}^D \mathbb{E}_q[t(x_i,z_i)] + \xi - \xi' \\ D + \nu - \nu' \end{bmatrix}$$

- Since preconditioning matrix is p.s.d., setting gradient to zero gives closed-form batch VB update

$$\xi' = \xi + \sum_{i=1}^D \mathbb{E}[t(x_i,z_i)], \qquad \nu' = \nu + D$$

- The SVI uses the stochastic gradients and the updates will be $\psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$

$$\begin{bmatrix} \xi'_t \\ \nu'_t \end{bmatrix} = \begin{bmatrix} \xi'_{t-1} \\ \nu'_{t-1} \end{bmatrix} - \rho_t M_t \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi',\nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'\partial\xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial\nu'^2} \end{bmatrix} \begin{bmatrix} \frac{D}{|S_t|}\sum_{i\in S_t} \mathbb{E}[t(x_i,z_i)] + \xi - \xi'_{t-1} \\ D + \nu - \nu'_{t-1} \end{bmatrix}$$

SVI for Exponential Family Distributions

- ELBO gradient (batch case)

$$\nabla_{(\xi',\nu')}\mathcal{L}_\beta = \begin{bmatrix} \nabla_{\xi'}\mathcal{L}_\beta \\ \frac{\partial}{\partial \nu'}\mathcal{L}_\beta \end{bmatrix} = -\begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi',\nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix} \begin{bmatrix} \sum_{i=1}^D \mathbb{E}_q[t(x_i,z_i)] + \xi - \xi' \\ D + \nu - \nu' \end{bmatrix}$$

- Since preconditioning matrix is p.s.d., setting gradient to zero gives closed-form batch VB update

$$\xi' = \xi + \sum_{i=1}^D \mathbb{E}[t(x_i,z_i)], \qquad \nu' = \nu + D$$

- The SVI uses the stochastic gradients and the updates will be $\psi_t = \psi_{t-1} + \rho_t M_t \nabla_\psi \mathcal{L}_t$

$$\begin{bmatrix} \xi'_t \\ \nu'_t \end{bmatrix} = \begin{bmatrix} \xi'_{t-1} \\ \nu'_{t-1} \end{bmatrix} - \rho_t M_t \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi',\nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix} \begin{bmatrix} \frac{D}{|S_t|}\sum_{i\in S_t} \mathbb{E}[t(x_i,z_i)] + \xi - \xi'_{t-1} \\ D + \nu - \nu'_{t-1} \end{bmatrix}$$

- M_t can be set to $\mathbf{I}$. However we can choose M_t sensibly to get a clean update.

SVI for Exponential Family Distributions

- Our stochastic gradient updates had the form

$$\begin{bmatrix} \xi'_t \\ \nu'_t \end{bmatrix} = \begin{bmatrix} \xi'_{t-1} \\ \nu'_{t-1} \end{bmatrix} - \rho_t M_t \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi', \nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix} \begin{bmatrix} \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}[t(x_i, z_i)] + \xi - \xi'_{t-1} \\ D + \nu - \nu'_{t-1} \end{bmatrix}$$

SVI for Exponential Family Distributions

- Our stochastic gradient updates had the form

$$\begin{bmatrix} \xi'_t \\ \nu'_t \end{bmatrix} = \begin{bmatrix} \xi'_{t-1} \\ \nu'_{t-1} \end{bmatrix} - \rho_t M_t \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi', \nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix} \begin{bmatrix} \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}[t(x_i, z_i)] + \xi - \xi'_{t-1} \\ D + \nu - \nu'_{t-1} \end{bmatrix}$$

- Suppose we set M_t as

$$M_t = - \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi', \nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix}^{-1}$$

SVI for Exponential Family Distributions

- Our stochastic gradient updates had the form

$$\begin{bmatrix} \xi'_t \\ \nu'_t \end{bmatrix} = \begin{bmatrix} \xi'_{t-1} \\ \nu'_{t-1} \end{bmatrix} - \rho_t M_t \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi', \nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix} \begin{bmatrix} \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}[t(x_i, z_i)] + \xi - \xi'_{t-1} \\ D + \nu - \nu'_{t-1} \end{bmatrix}$$

- Suppose we set M_t as

$$M_t = - \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi', \nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix}^{-1}$$

- Then the updates will be

$$\xi'_t = (1 - \rho_t)\xi'_{t-1} + \rho_t \left(\xi + \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}[t(x_i, z_i)] \right) \qquad \nu'_t = (1 - \rho_t)\nu'_{t-1} + \rho_t(\nu + D)$$

SVI for Exponential Family Distributions

- Our stochastic gradient updates had the form

$$\begin{bmatrix} \xi'_t \\ \nu'_t \end{bmatrix} = \begin{bmatrix} \xi'_{t-1} \\ \nu'_{t-1} \end{bmatrix} - \rho_t M_t \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi', \nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix} \begin{bmatrix} \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}[t(x_i, z_i)] + \xi - \xi'_{t-1} \\ D + \nu - \nu'_{t-1} \end{bmatrix}$$

- Suppose we set M_t as

$$M_t = - \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi', \nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix}^{-1}$$

- Then the updates will be

$$\xi'_t = (1 - \rho_t)\xi'_{t-1} + \rho_t \left(\xi + \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}[t(x_i, z_i)] \right) \qquad \nu'_t = (1 - \rho_t)\nu'_{t-1} + \rho_t(\nu + D)$$

- Note: The above choice of M_t is not arbitrary. It is actually equivalent to $M_t = -\mathbb{E}_q[\nabla^2 \ln q(\beta)]$

SVI for Exponential Family Distributions

- Our stochastic gradient updates had the form

$$\begin{bmatrix} \xi'_t \\ \nu'_t \end{bmatrix} = \begin{bmatrix} \xi'_{t-1} \\ \nu'_{t-1} \end{bmatrix} - \rho_t M_t \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi', \nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix} \begin{bmatrix} \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}[t(x_i, z_i)] + \xi - \xi'_{t-1} \\ D + \nu - \nu'_{t-1} \end{bmatrix}$$

- Suppose we set M_t as

$$M_t = - \begin{bmatrix} \nabla^2_{\xi'} \ln f(\xi', \nu') & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi} \\ \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu' \partial \xi'^T} & \frac{\partial^2 \ln f(\xi',\nu')}{\partial \nu'^2} \end{bmatrix}^{-1}$$

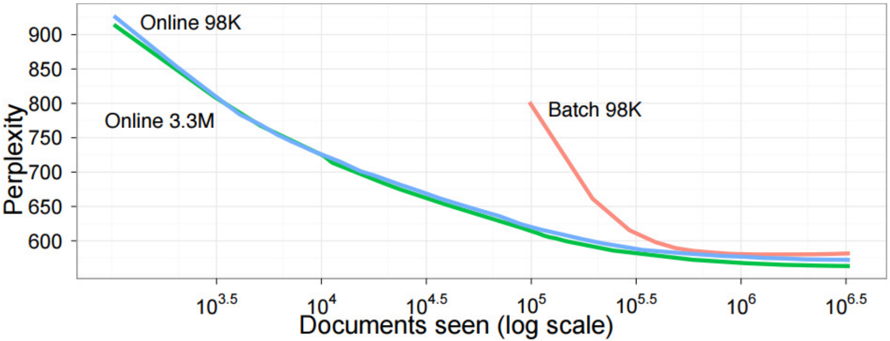
- Then the updates will be

$$\xi'_t = (1 - \rho_t)\xi'_{t-1} + \rho_t \left(\xi + \frac{D}{|S_t|} \sum_{i \in S_t} \mathbb{E}[t(x_i, z_i)] \right) \qquad \nu'_t = (1 - \rho_t)\nu'_{t-1} + \rho_t(\nu + D)$$

- Note: The above choice of M_t is not arbitrary. It is actually equivalent to $M_t = -\mathbb{E}_q[\nabla^2 \ln q(\beta)]$
  - This choice makes our stochastic gradient a "natural gradient" (considered to be good direction)

SVI vs Batch Inference

- Online inference methods (e.g., SVI) usually have a faster convergence than batch inference
- Shown below is a plot comparing batch and online inference for topic model (LDA)



(Pic courtesy: David Blei)

Summary

- VB is a deterministic approximate inference method (unlike sampling methods)

Summary

- VB is a deterministic approximate inference method (unlike sampling methods)
- Finds the best q by maximizing the ELBO (or minimizing $\text{KL}(q||p)$)

Summary

- VB is a deterministic approximate inference method (unlike sampling methods)
- Finds the best q by maximizing the ELBO (or minimizing $\text{KL}(q||p)$)
- VB is guaranteed to converge to a local optima (in finite time)

Summary

- VB is a deterministic approximate inference method (unlike sampling methods)
- Finds the best q by maximizing the ELBO (or minimizing $\text{KL}(q||p)$)
- VB is guaranteed to converge to a local optima (in finite time)
- Simplifying assumption on q (e.g., mean field VB) can make VB updates very easy to derive

Summary

- VB is a deterministic approximate inference method (unlike sampling methods)
- Finds the best q by maximizing the ELBO (or minimizing $\text{KL}(q||p)$)
- VB is guaranteed to converge to a local optima (in finite time)
- Simplifying assumption on q (e.g., mean field VB) can make VB updates very easy to derive
- More advanced VB methods can handle richer forms of $q(\mathbf{Z})$, likelihood and priors

Summary

- VB is a deterministic approximate inference method (unlike sampling methods)
- Finds the best q by maximizing the ELBO (or minimizing $\text{KL}(q||p)$)
- VB is guaranteed to converge to a local optima (in finite time)
- Simplifying assumption on q (e.g., mean field VB) can make VB updates very easy to derive
- More advanced VB methods can handle richer forms of $q(\mathbf{Z})$, likelihood and priors
    - E.g., Monte-Carlo approximations of ELBO and its derivatives

Summary

- VB is a deterministic approximate inference method (unlike sampling methods)
- Finds the best q by maximizing the ELBO (or minimizing $\text{KL}(q||p)$)
- VB is guaranteed to converge to a local optima (in finite time)
- Simplifying assumption on q (e.g., mean field VB) can make VB updates very easy to derive
- More advanced VB methods can handle richer forms of $q(\mathbf{Z})$, likelihood and priors
  - E.g., Monte-Carlo approximations of ELBO and its derivatives
- Combination of methods like BBVI and SVI can help us develop scalable approximate inference algorithms for a large class of models

Summary

- VB is a deterministic approximate inference method (unlike sampling methods)
- Finds the best q by maximizing the ELBO (or minimizing $\mathrm{KL}(q||p)$)
- VB is guaranteed to converge to a local optima (in finite time)
- Simplifying assumption on q (e.g., mean field VB) can make VB updates very easy to derive
- More advanced VB methods can handle richer forms of $q(\mathbf{Z})$, likelihood and priors
  - E.g., Monte-Carlo approximations of ELBO and its derivatives
- Combination of methods like BBVI and SVI can help us develop scalable approximate inference algorithms for a large class of models
- Implementations of many classic/advanced VB methods available in Stan, Edward, etc.