Mean Field (Cont'd) + Stochastic Gradient Descend, SVI, and Scalability

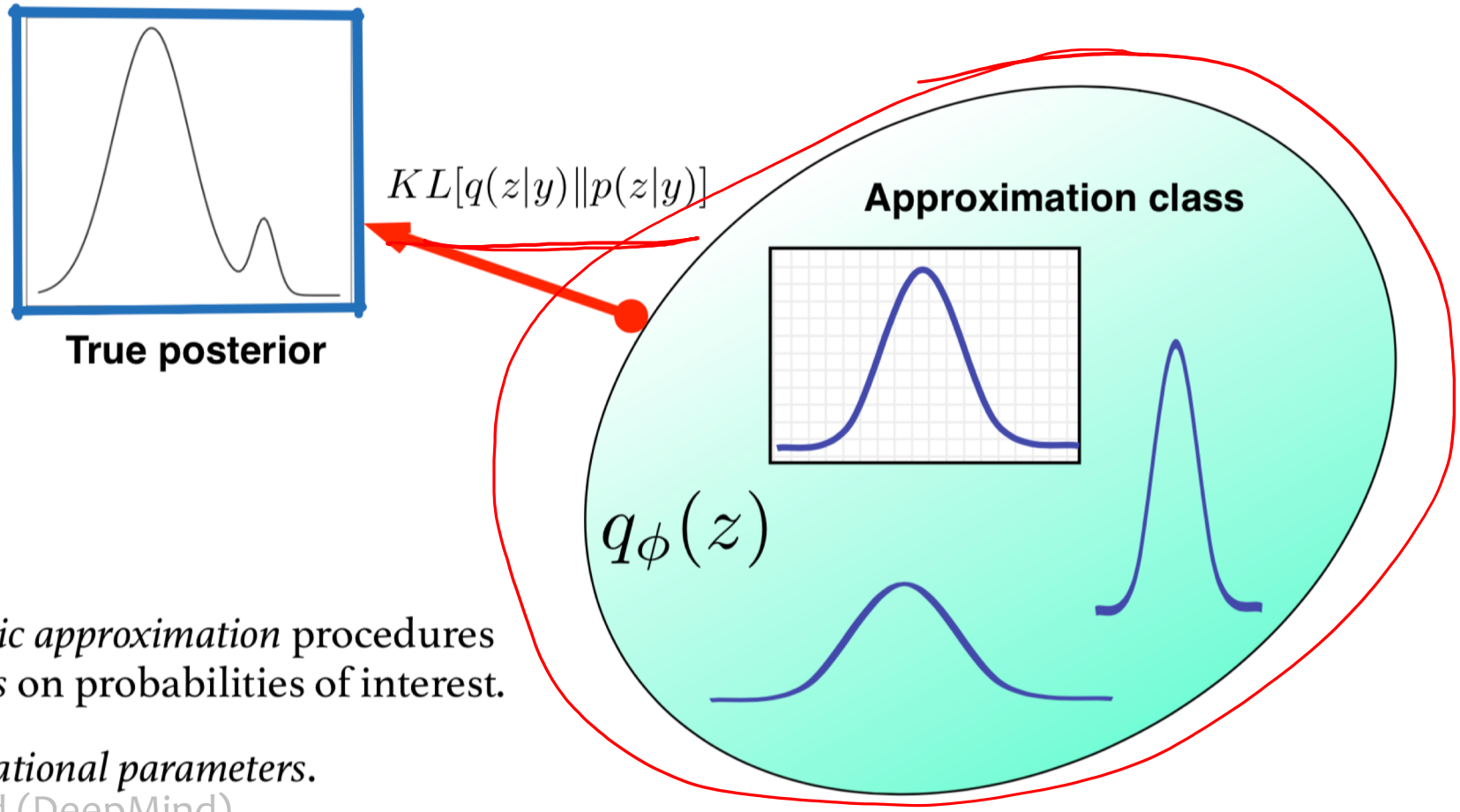
Kayhan Batmanghelich

Slides Credit (Partially adopted from):

- **Shakir Mohamed (DeepMind)**

Variational Methods

Variational Principle
General family of methods for approximating complicated densities by a simpler class of densities.



Deterministic *approximation* procedures with *bounds* on probabilities of interest.

Fit the *variational parameters*.

Slides Credit: Shakir Mohamed (DeepMind)

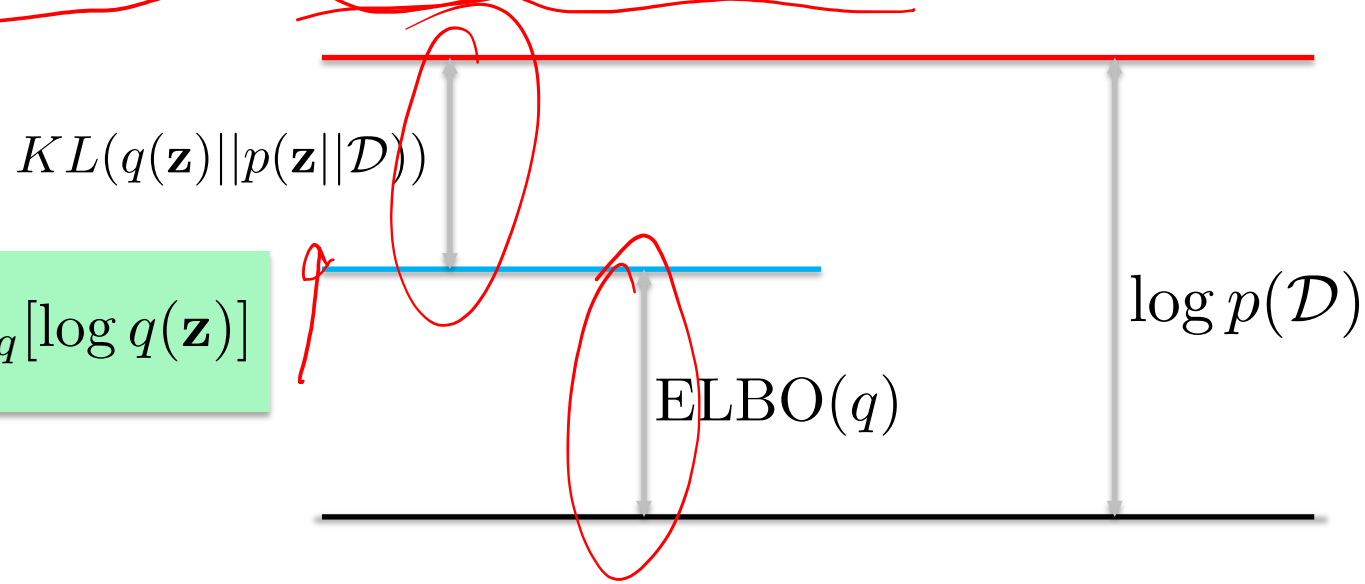
Recap: Evidence Lower Bound (ELBO)

$$q^*(\mathbf{z}) = \arg\min_{q(\mathbf{z})\in\mathcal{Q}} KL(q(\mathbf{z})||p(\mathbf{z}||\mathcal{D})) \rightarrow \mathbb{E}_q\left[\log \frac{q(z)}{p(z|D)}\right]$$

$$KL(q(\mathbf{z})||p(\mathbf{z}||\mathcal{D})) = \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z}|\mathcal{D})]$$

$$KL(q(\mathbf{z})||p(\mathbf{z}||\mathcal{D})) = \mathbb{E}_q[\log q(\mathbf{z})] - \mathbb{E}_q[\log p(\mathbf{z},\mathcal{D})] + \log p(\mathcal{D})$$

$$\text{ELBO}(q) = \mathbb{E}_q[\log p(\mathbf{z},\mathcal{D})] - \mathbb{E}_q[\log q(\mathbf{z})]$$



Interpreting the Lower Bound (ELBO)

$$\mathcal{F}(y, q) = \mathbb{E}_{q(z)}\left[\log p(y|z)\right] - KL\left[q(z)||p(z)\right]$$

Data

Approximate Posterior

Reconstruction

Penalty

Approximate posterior distribution *q(z)*: Best match to true posterior *p(z|y)*, one of the unknown inferential quantities of interest to us.

Reconstruction Cost: The expected log-likelihood measure how well samples from *q(z)* are able to explain the data *y*.

Penalty: Ensures the the explanation of the data *q(z)* doesn't deviate too far from your beliefs *p(z)*. A mechanism for realising Okham's razor.

$$\mathcal{F}(y, q) = \mathbb{E}_{q(z)}\left[\log p(y|z)\right] - KL\left[q(z)||p(z)\right]$$

Approximate Posterior

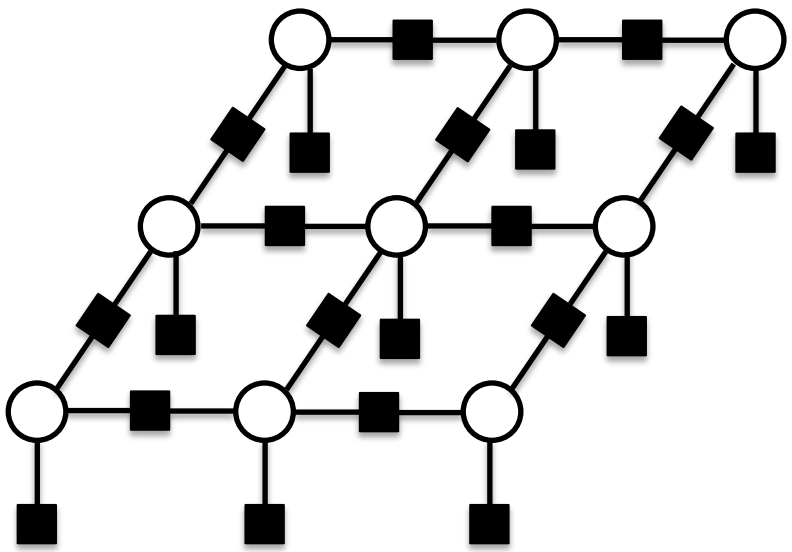
How to implement it?
What is q exactly?

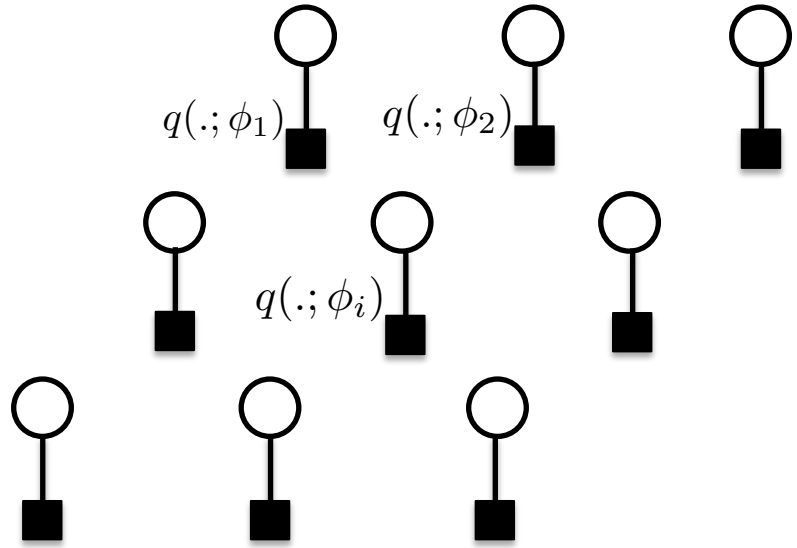
Recap: (Naïve) Mean Field Approach

Very popular approach assuming the posterior is fully factorizable

$$q(\mathbf{x}; \boldsymbol{\phi}) = \prod_i q_i(x_i; \phi_i)$$

$q_1 \times q_2 - - - -$





Recap: Mean Field Updates

q_1 q_k

$p(z|D) = \frac{p(z,D)}{p(D)}$

$p(z|D) \propto p(z,D)$

Let's focus on q_j (holding all other terms constant)

$p(z,D)$

$$L(q_j) = \sum_{\mathbf{x}} \prod_i q_i(\mathbf{x}) \left[\log \tilde{p}(\mathbf{x}) - \sum_k \log q_k(\mathbf{x}_k) \right]$$

$$= \sum_{\mathbf{x}_j} q_j(\mathbf{x}_j) \boxed{\log f_j(\mathbf{x}_j)} - \sum_{\mathbf{x}_j} q_j(\mathbf{x}_j) \log q_j(\mathbf{x}_j) + \text{const}$$

$$\sum_{\mathbf{x}_j} q_j(\mathbf{x}_j) \sum_{\mathbf{x}_{-j}} \prod_{i \neq j} q_i(\mathbf{x}_i) \log \tilde{p}(\mathbf{x})$$

$\mathbb{E}_{q_{-j}}[\ \ ]$

$$L(q_j) = \mathbb{E}_{q_j}\left[\mathbb{E}_{q_{-j}}[\log \tilde{p}(\mathbf{x})]\right] + H(q_j)$$

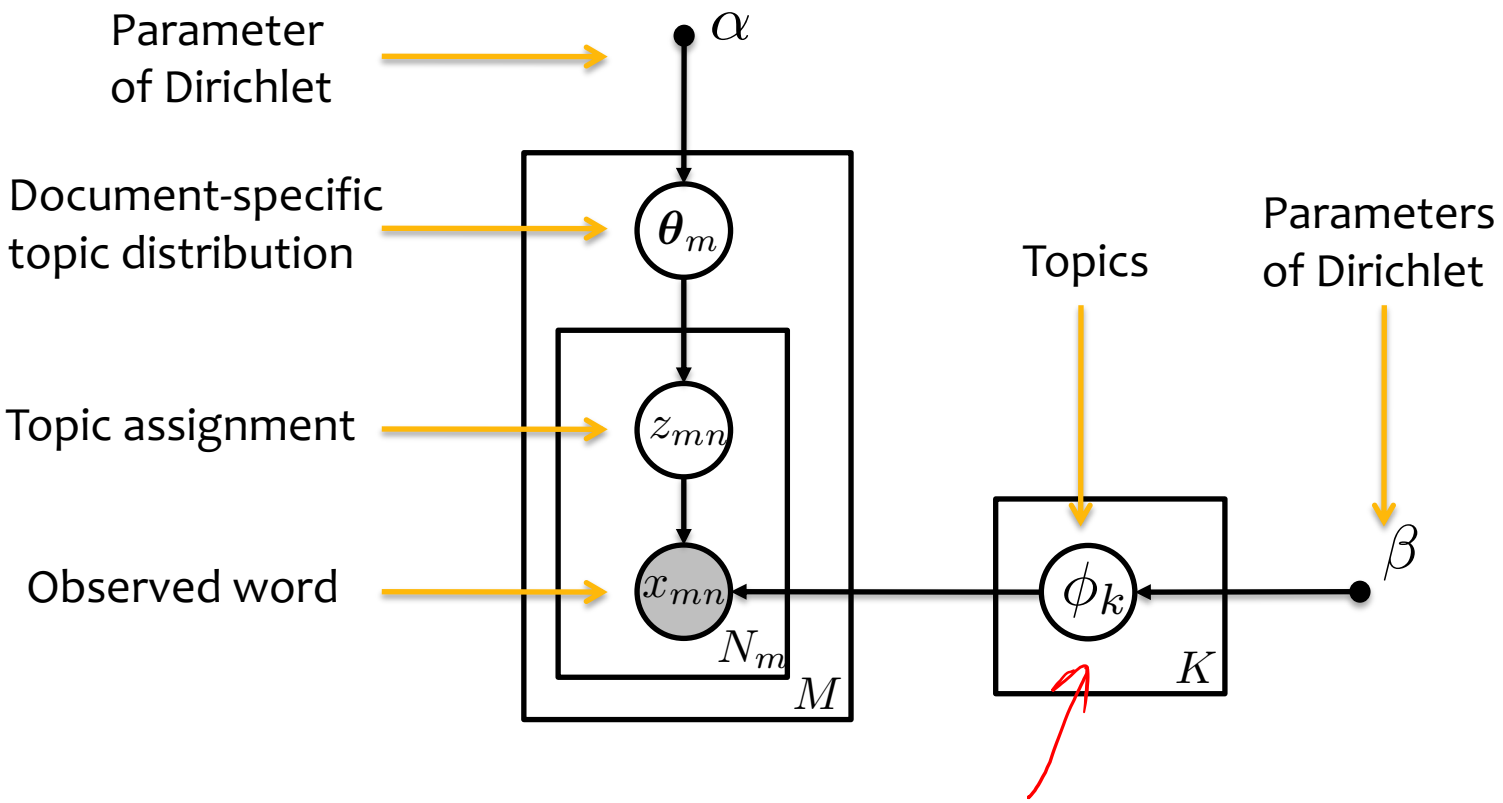
$$\frac{\delta L(q_j)}{\delta q_j} = 0 \qquad \frac{\delta L(q_j)}{\delta q_j} = \mathbb{E}_{q_{-j}}[\log \tilde{p}(\mathbf{x})] - \log q_j - 1 = 0$$

$$q_j^* \propto \exp\{\mathbb{E}_{q_{-j}}[\log \tilde{p}(\mathbf{x})]\}$$

Case study: Latent Dirichlet Allocation

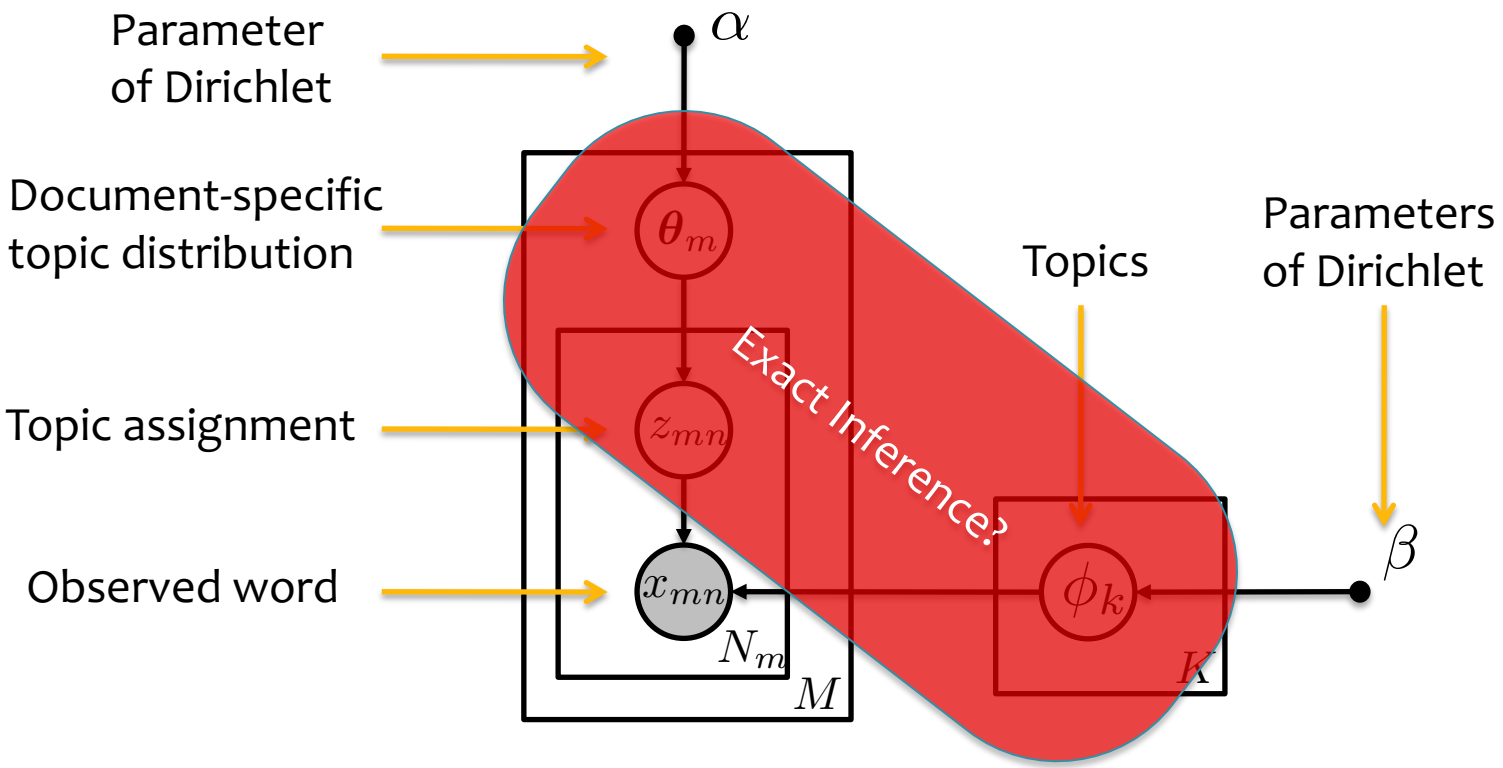
Latent Dirichlet Allocation

- Plate Diagram



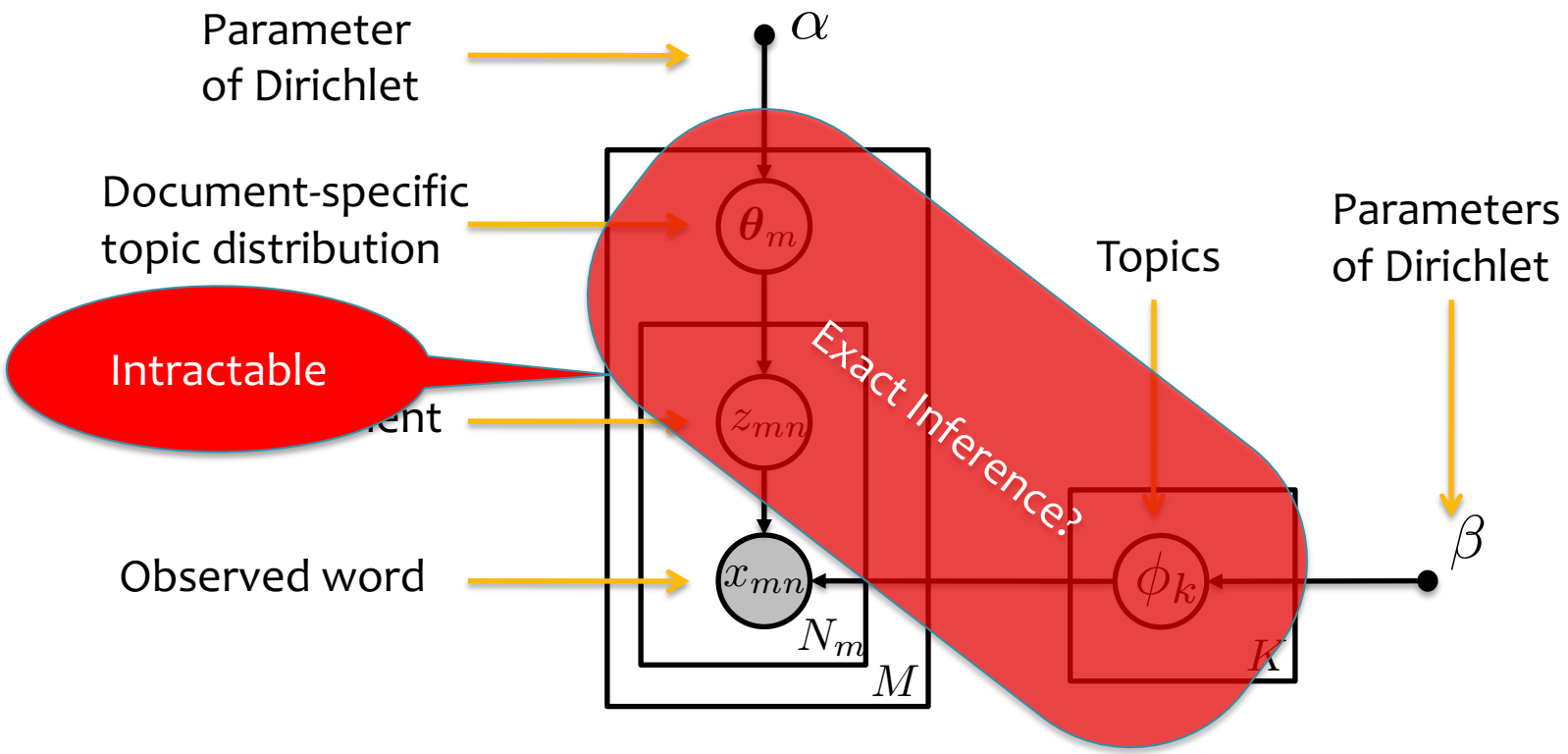
Latent Dirichlet Allocation

- Plate Diagram



Latent Dirichlet Allocation

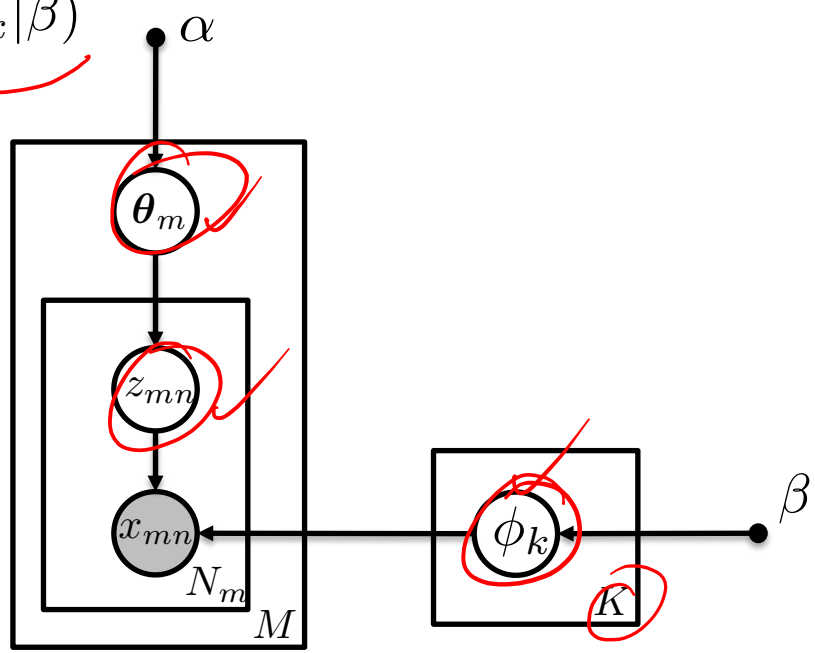
- Plate Diagram



Inference

- Joint distribution

$$p(\cdot) = \prod_{m}^{M} p(\theta_m|\alpha) \prod_{n=1}^{N_m} p(x_{mn}|z_{nm}, \{\phi_k\}_{k=1}^{K}) p(z_{nm}|\theta_m) \prod_{k}^{K} p(\phi_k|\beta)$$



- Latent variables

$$\{\phi_k\}_{k=1}^{K}, \{z_{nm}\}, \{\theta_m\}$$

- Posterior distribution

$$q(\cdot) = \prod_{k=1}^{K} p(\phi_k) \prod_{m=1} p(\theta_m) \prod_{n=1} p(z_{nm})$$

$q(\{\phi_k\}, \{z\}, \{\theta\})$

Let's work out one of the updates….

$\prod \theta_k^{1(z_n=k)}$

$$q(\theta_m) \propto \exp\left[\mathbb{E}_{\prod_n q(z_{nm})}[\log p(\theta_m|\alpha)] + \sum_n \log p(z_{nm}|\theta_m)\right]$$

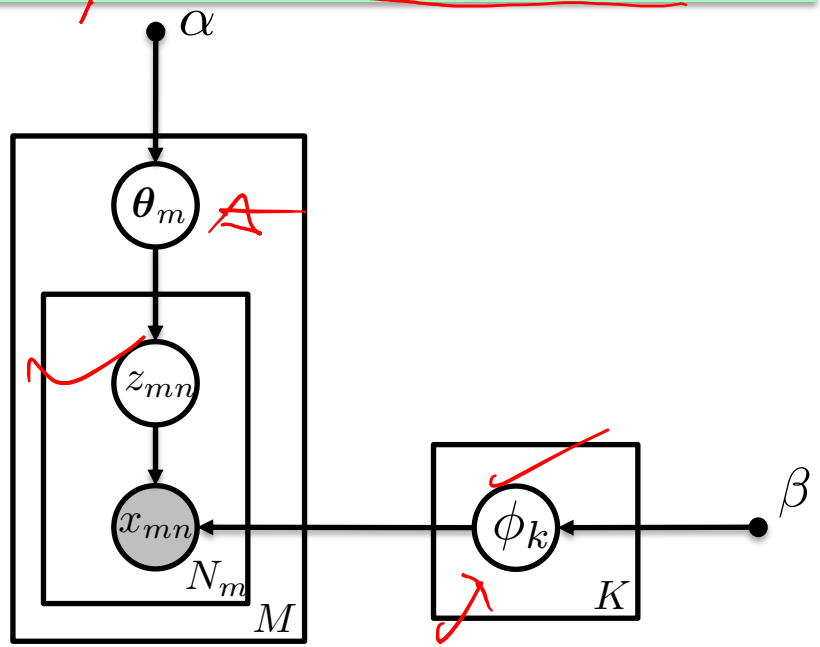
Remember this:

$$q_j^* \propto \exp\{\mathbb{E}_{q_{-j}}[\log \tilde{p}(\mathbf{x})]\}$$

In LDA:

Dirichlet: $$p(\theta_m|\alpha) \propto \exp\left[\sum_{k=1}^{K}(\alpha_k - 1)\log\theta_{mk}\right]$$

Categorical: $$p(z_{mn}|\theta_m) \propto \exp\left[\sum_{k=1}^{K} 1(z_{mn}=k)\log\theta_{mk}\right]$$



We Obtain:

$$q(\theta_m) \propto \exp\left[\sum_{k=1}^{K}\left(\sum_{n=1}^{N} q(z_{mn}=k) + \alpha_k - 1\right)\log\theta_{mn}\right]$$

The posterior is a Dirichlet

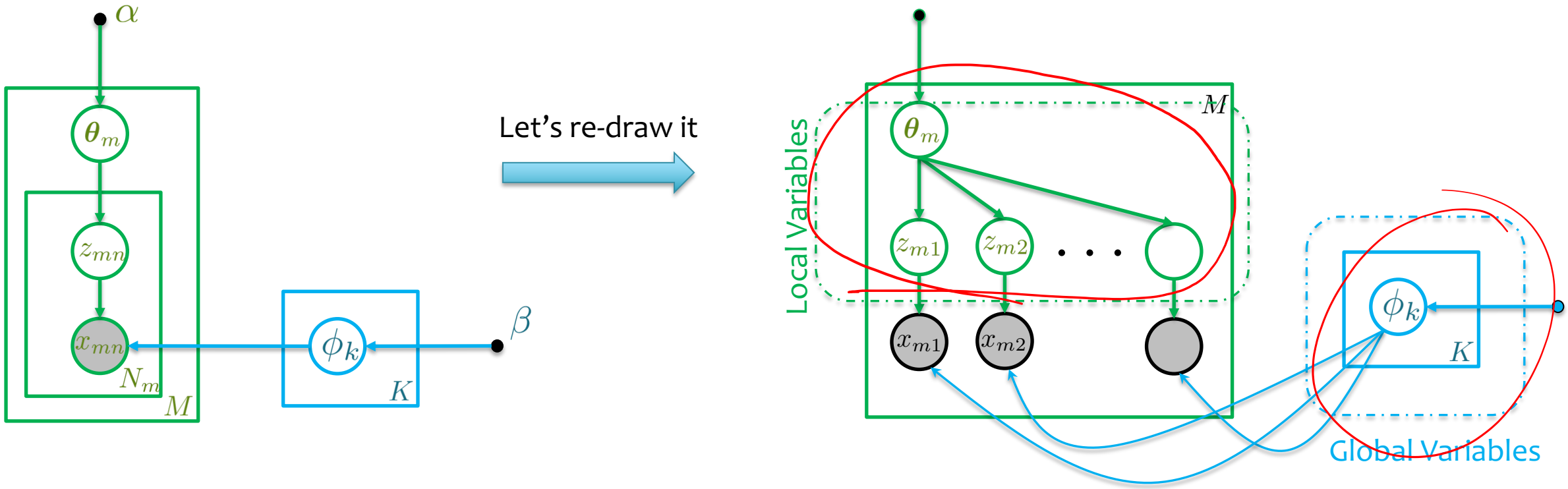
Why? $E_{x,y,z}[f(x,y,z)] = E_x E_y E_z[f(\cdot)]$

Dirichlet-Categorical Model

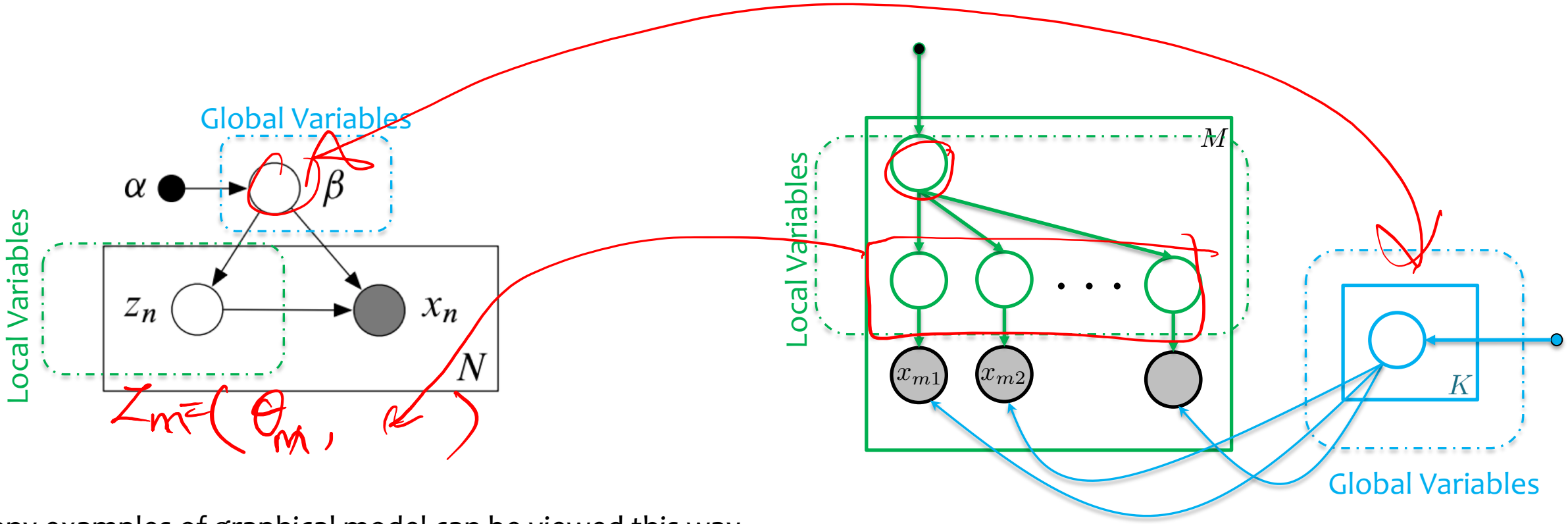
- Why conjugacy is so useful

$$
\begin{aligned}
p(X|\boldsymbol{\alpha}) &= \int_{\phi} p(X|\vec{\phi})p(\vec{\phi}|\boldsymbol{\alpha})\, d\phi \\
&= \int_{\phi} \left(\prod_{v=1}^{V} \phi_v^{n_v}\right)\left(\frac{1}{B(\boldsymbol{\alpha})}\prod_{v=1}^{V} \phi_v^{\alpha_v - 1}\right) d\phi \\
&= \frac{1}{B(\boldsymbol{\alpha})}\int_{\phi} \prod_{v=1}^{V} \phi_v^{n_v+\alpha_v-1}\, d\phi \\
&= \frac{1}{B(\boldsymbol{\alpha})}\int_{\phi} \frac{B(\vec{n}+\boldsymbol{\alpha})}{B(\vec{n}+\boldsymbol{\alpha})}\prod_{v=1}^{V} \phi_v^{n_v+\alpha_v-1}\, d\phi \\
&= \frac{B(\vec{n}+\boldsymbol{\alpha})}{B(\boldsymbol{\alpha})}\int_{\phi} \underbrace{\frac{1}{B(\vec{n}+\boldsymbol{\alpha})}\prod_{v=1}^{V} \phi_v^{n_v+\alpha_v-1}}_{Dir(\vec{n}+\boldsymbol{\alpha})}\, d\phi \\
&= \frac{B(\vec{n}+\boldsymbol{\alpha})}{B(\boldsymbol{\alpha})}
\end{aligned}
$$

Let's generalize that example….

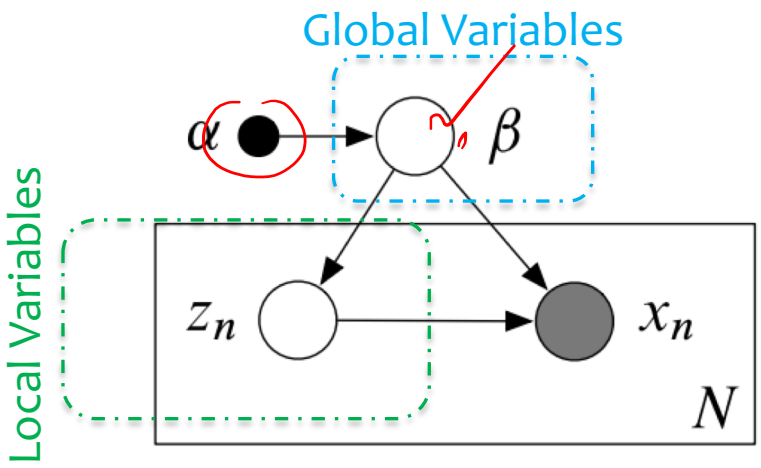


Let's generalize that example….



Many examples of graphical model can be viewed this way

Let's generalize that example….



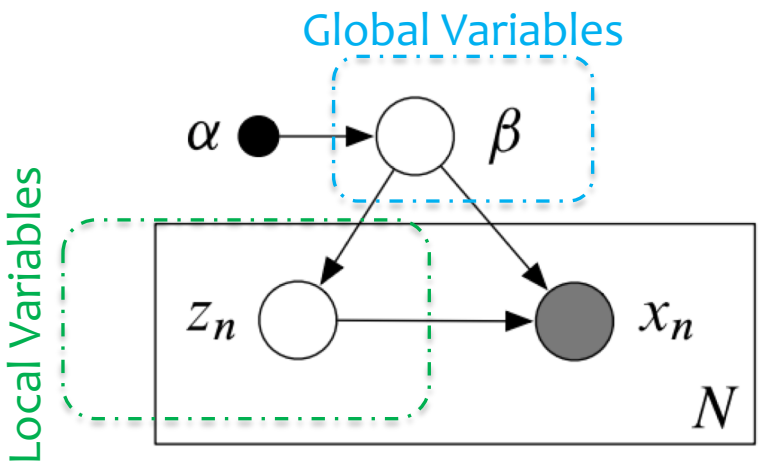
A few observations:

$$p(x, z, \beta \mid \alpha) = p(\beta \mid \alpha) \prod_{n=1}^{N} p(x_n, z_n \mid \beta).$$

$$p(x_n, z_n \mid \cancel{x_{-n}, z_{-n}}, \beta, \alpha) = p(x_n, z_n \mid \beta, \alpha).$$

M.D. Hoffman et al., "Stochastic Variational Inference," https://arxiv.org/abs/1206.7051

Let's generalize that example….



A few observations:

$$p(x, z, \beta \mid \alpha) = p(\beta \mid \alpha) \prod_{n=1}^{N} p(x_n, z_n \mid \beta).$$

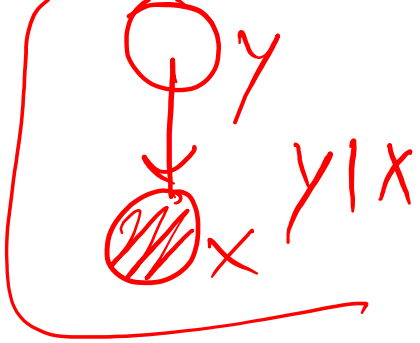
$$p(x_n, z_n \mid x_{-n}, z_{-n}, \beta, \alpha) = p(x_n, z_n \mid \beta, \alpha).$$

Further assumptions: (1) Exponential family, (2) conjugacy

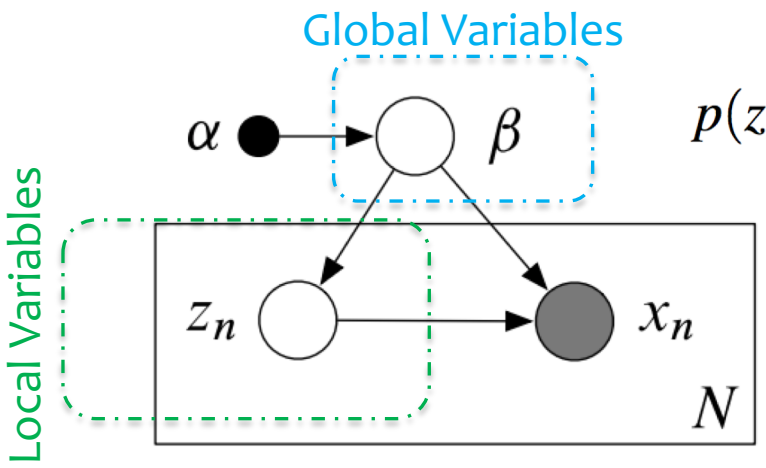
$$p(\beta \mid x, z, \alpha) = h(\beta) \exp\{\eta_g(x, z, \alpha)^\top t(\beta) - a_g(\eta_g(x, z, \alpha))\},$$

$$p(z_{nj} \mid x_n, z_{n,-j}, \beta) = h(z_{nj}) \exp\{\eta_\ell(x_n, z_{n,-j}, \beta)^\top t(z_{nj}) - a_\ell(\eta_\ell(x_n, z_{n,-j}, \beta))\}.$$

M.D. Hoffman et al., "Stochastic Variational Inference," https://arxiv.org/abs/1206.7051



Let's generalize that example….



Model:

$$p(\beta \mid x, z, \alpha) = h(\beta) \exp\{\eta_g(x, z, \alpha)^\top t(\beta) - a_g(\eta_g(x, z, \alpha))\},$$

$$p(z_{nj} \mid x_n, z_{n,-j}, \beta) = h(z_{nj}) \exp\{\eta_\ell(x_n, z_{n,-j}, \beta)^\top t(z_{nj}) - a_\ell(\eta_\ell(x_n, z_{n,-j}, \beta))\}.$$

Approximate Posterior (mean field):

$$q(z, \beta) = \boxed{q(\beta \mid \lambda)} \prod_{n=1}^{N} \prod_{j=1}^{J} q(z_{nj} \mid \phi_{nj}).$$

$$q(\beta \mid \lambda) = h(\beta) \exp\{\lambda^\top t(\beta) - a_g(\lambda)\},$$

$$q(z_{nj} \mid \phi_{nj}) = h(z_{nj}) \exp\{\phi_{nj}^\top t(z_{nj}) - a_\ell(\phi_{nj})\}.$$

$$q_j^* \propto \exp\{\mathbb{E}_{q_{-j}}[\log \tilde{p}(\mathbf{x})]\}$$

M.D. Hoffman et al., "Stochastic Variational Inference," https://arxiv.org/abs/1206.7051

Let's generalize that example….

Model:

$$p(\beta \mid x, z, \alpha) = h(\beta) \exp\{\eta_g(x, z, \alpha)^\top t(\beta) - a_g(\eta_g(x, z, \alpha))\},$$

$$p(z_{nj} \mid x_n, z_{n,-j}, \beta) = h(z_{nj}) \exp\{\eta_\ell(x_n, z_{n,-j}, \beta)^\top t(z_{nj}) - a_\ell(\eta_\ell(x_n, z_{n,-j}, \beta))\}.$$

Approximate Posterior (mean field):

$$q(z, \beta) = q(\beta \mid \lambda) \prod_{n=1}^{N} \prod_{j=1}^{J} q(z_{nj} \mid \phi_{nj}).$$

1: Initialize $\lambda^{(0)}$ randomly.
2: **repeat**
3: **for** each local variational parameter ϕ_{nj} **do**
4: Update ϕ_{nj}, $\phi_{nj}^{(t)} = \mathbb{E}_{q^{(t-1)}}[\eta_{\ell,j}(x_n, z_{n,-j}, \beta)]$.
5: **end for**
6: Update the global variational parameters, $\lambda^{(t)} = \mathbb{E}_{q^{(t)}}[\eta_g(z_{1:N}, x_{1:N})]$.
7: **until** the ELBO converges

M.D. Hoffman et al., "Stochastic Variational Inference," https://arxiv.org/abs/1206.7051

Optimization View

Decoder Encoder View

$$\max_{\phi,\theta} \mathcal{F}(y, q_\phi) = \mathbb{E}_{q_\phi(z)}\left[\log p_\theta(y|z)\right] - KL\left[q_\phi(z)||p(z)\right]$$

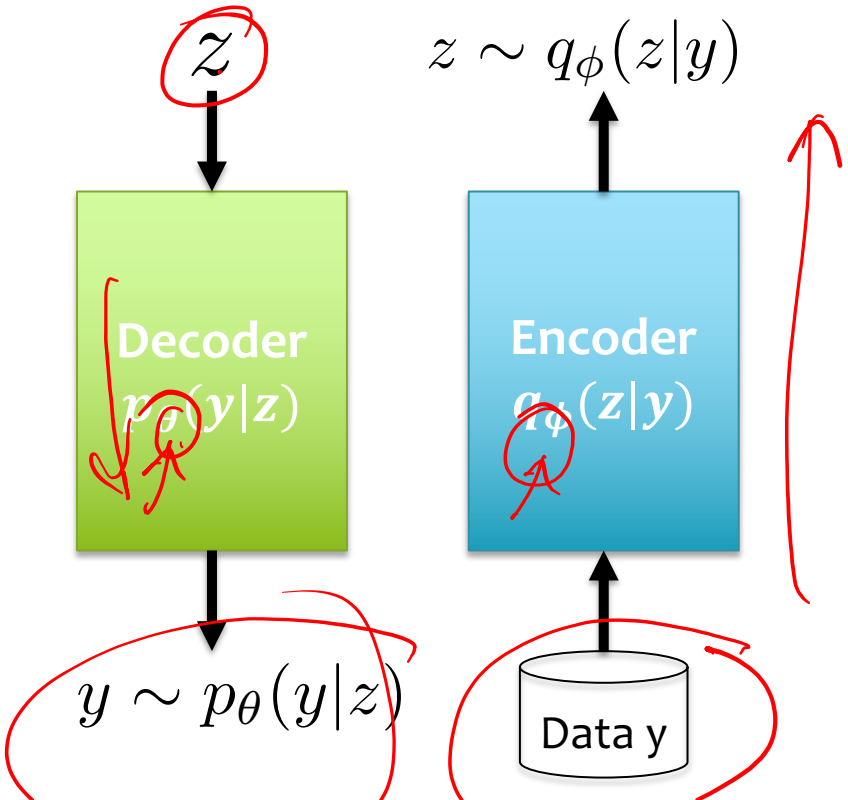
Stochastic encoder

Data code-length

Hypothesis code

Encoder: variational distribution $q_\phi(z|y)$

Decoder: likelihood $p_\theta(y|z)$



Slides Credit: Shakir Mohamed (DeepMind)

Variational EM

$$\max_{\phi,\theta} \mathcal{F}(y, q_\phi) = \mathbb{E}_{q_\phi(z)}\left[\log p_\theta(y|z)\right] - KL\left[q_\phi(z)||p(z)\right]$$

Alternative optimization for the variational parameters and then model parameters (VEM).

Repeat:

E-Step: **(inference)**

For i=1,…,N

How to compute these?

$$\phi_n \propto \nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(y_n|z_n)] - \nabla_\phi KL[q(z_n)||p(z_n)]$$

M-Step: **(Parameter Learning)**

$$\theta \propto \frac{1}{N}\sum_n \mathbb{E}_{q_\phi(z)}\left[\nabla_\theta \log p_\theta(y_n|z_n)\right]$$

We know how to estimate

Slides Credit: Shakir Mohamed (DeepMind)

Monte Carlo Gradient Approximation

White board: How to compute

$$\nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(y_n|z_n)]$$

$$\nabla_\phi \mathbb{E}_{q_\phi(z)}[f(z)]$$

$$\nabla \log q_\phi(z)$$

$$\nabla_\phi \int f(z)\, q_\phi(z)\, dz$$

$$\int f(z) \nabla q_\phi(z)\, dz$$

$$\int f(z) \frac{\nabla_\phi q(z)}{q_\phi(z)} q(z)\, dz$$

$$\mathbb{E}_{q_\phi(z)}[f(z)\nabla_\phi \log q_\phi(z)]$$

Controlling the Variance

A.C. Miller et al., "Reducing Reparameterization Gradient Variance," https://arxiv.org/pdf/1705.07880.pdf

The general idea:

Control variates: Control variates are random quantities that are used to **reduce the variance** of a statistical estimator **without introducing any bias** by injecting information into the estimator. Let's say you want to estimate $\mathbb{E}[\hat{g}] = g$, the idea is to use **another random variable $\tilde{g}$** with known mean $\tilde{m}$ that is correlated with $\hat{g}$

No bias was introduced

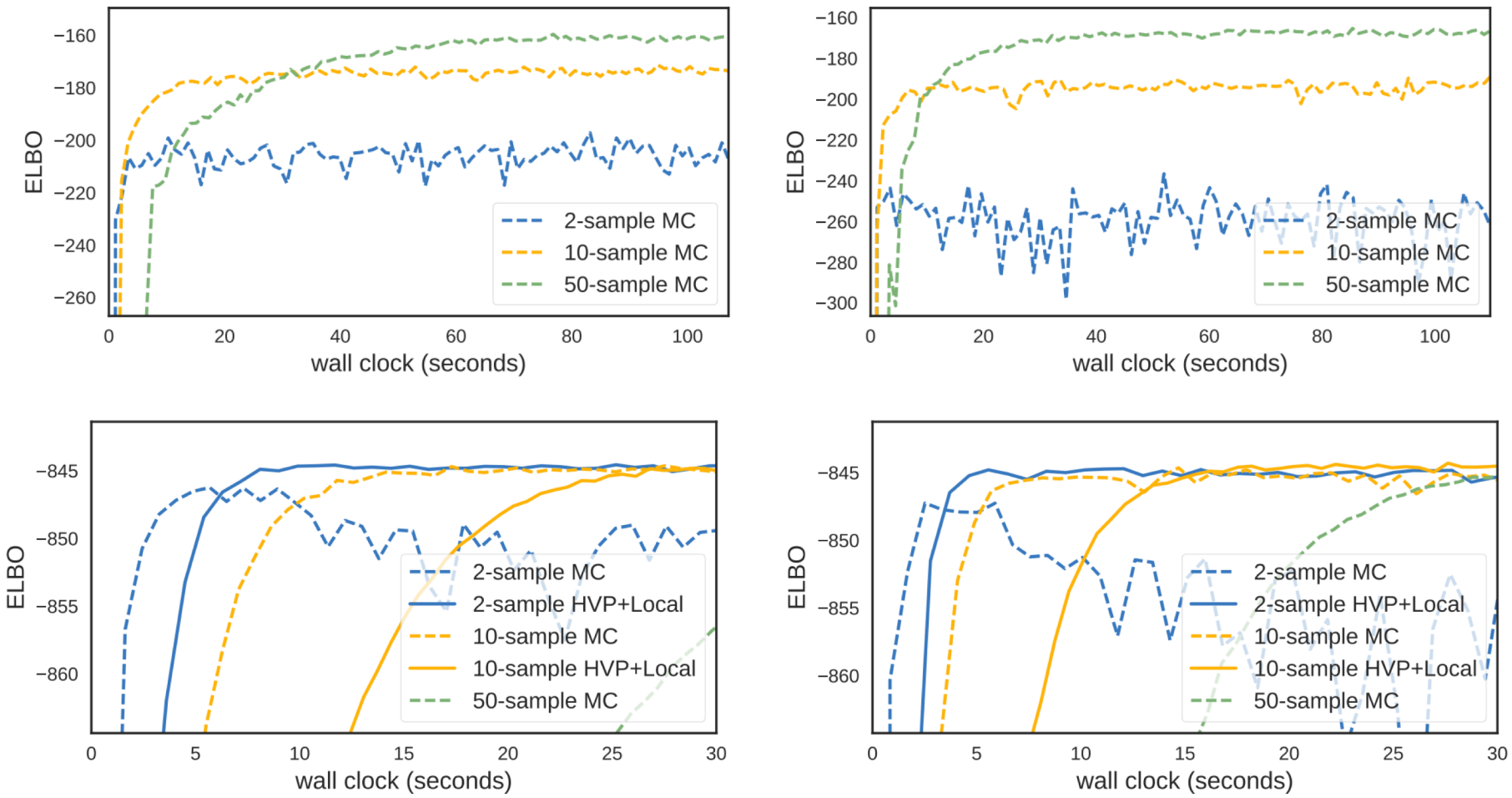
$$\boldsymbol{g}^{(cv)} = \hat{\boldsymbol{g}} - c \cdot (\tilde{\boldsymbol{g}} - \tilde{\boldsymbol{m}}) \,.$$

$$\begin{aligned}
\mathbb{V}(\boldsymbol{g}^{(cv)}) &= \mathbb{E}[(\hat{\boldsymbol{g}} - c \cdot \tilde{\boldsymbol{g}})^2] - \mathbb{E}[\hat{\boldsymbol{g}}]^2 \\
&= \mathbb{E}[\hat{\boldsymbol{g}}^2 + c^2 \cdot \tilde{\boldsymbol{g}}^2 - 2c\hat{\boldsymbol{g}}\tilde{\boldsymbol{g}}] - \mathbb{E}[\hat{\boldsymbol{g}}]^2 \\
&= \mathbb{E}[\hat{\boldsymbol{g}}^2] + c^2\mathbb{E}[\tilde{\boldsymbol{g}}^2] - 2c\mathbb{E}[\hat{\boldsymbol{g}}\tilde{\boldsymbol{g}}] - \mathbb{E}[\hat{\boldsymbol{g}}]^2
\end{aligned}$$

$$c^* = \frac{\mathbb{E}[\hat{\boldsymbol{g}}\tilde{\boldsymbol{g}}]}{\mathbb{E}[\tilde{\boldsymbol{g}}^2]} = \frac{\mathbb{C}(\hat{\boldsymbol{g}}, \tilde{\boldsymbol{g}})}{\mathbb{V}(\tilde{\boldsymbol{g}})} \quad \Rightarrow \quad \mathbb{V}(\boldsymbol{g}^{(cv)}) = (1 - \rho^2)\mathbb{V}(\hat{\boldsymbol{g}})$$

Controlling the Variance

A.C. Miller et al., "Reducing Reparameterization Gradient Variance," https://arxiv.org/pdf/1705.07880.pdf



(a) adam with step size = 0.05

(b) adam with step size = .10

On scalability of Variational EM

$$\max_{\phi,\theta} \mathcal{F}(y, q_\phi) = \mathbb{E}_{q_\phi(z)}\left[\log p_\theta(y|z)\right] - KL\left[q_\phi(z)||p(z)\right]$$

Repeat:

E-Step: **(inference)**

For i=1,…,N

$$\phi_n \propto \nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(y_n|z_n)] - \nabla_\phi KL[q(z_n)||p(z_n)]$$

M-Step: **(Parameter Learning)**

$$\theta \propto \boxed{\frac{1}{N}\sum_n} \mathbb{E}_{q_\phi(z)}\left[\nabla_\theta \log p_\theta(y_n|z_n)\right]$$

Slides Credit: Shakir Mohamed (DeepMind)

On scalability of Variational EM

$$\max_{\phi,\theta} \mathcal{F}(y, q_\phi) = \mathbb{E}_{q_\phi(z)}\left[\log p_\theta(y|z)\right] - KL\left[q_\phi(z)||p(z)\right]$$

Repeat:

E-Step: **(inference)**

For i=1,…,N

***N* is a mini-batch** - sampled with replacement from the full data set or received online.

$$\phi_n \propto \nabla_\phi \mathbb{E}_{q_\phi(z)}[\log p_\theta(y_n|z_n)] - \nabla_\phi KL[q(z_n)||p(z_n)]$$

M-Step: **(Parameter Learning)**

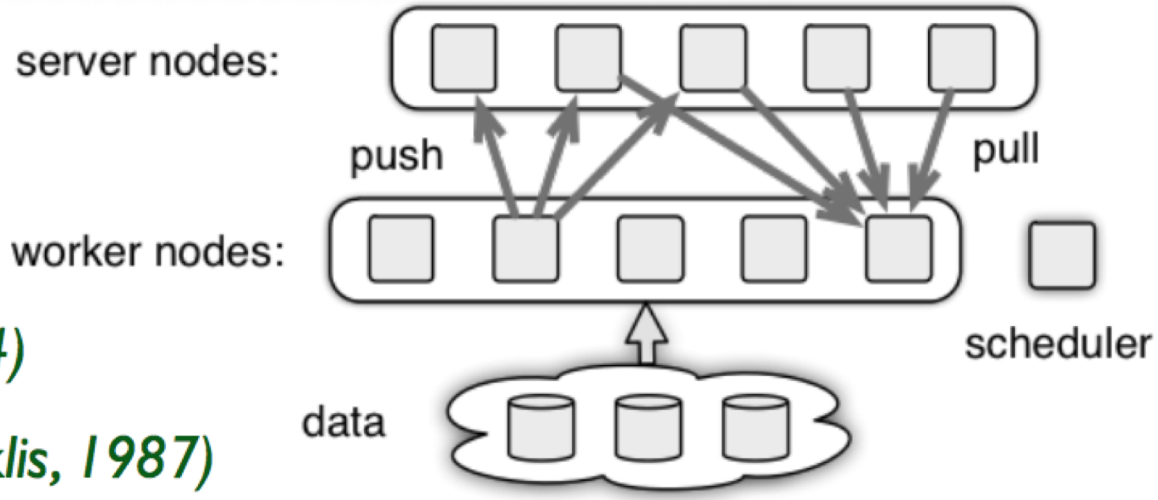
$$\theta \propto \boxed{\frac{1}{N}\sum_n} \mathbb{E}_{q_\phi(z)}\left[\nabla_\theta \log p_\theta(y_n|z_n)\right]$$

Scalable - only need to operate on a small batch at a time. Can operate on large data sets.

Slides Credit: Shakir Mohamed (DeepMind)

Parallelization

common parameter server architecture



(Li, Andersen, Smola, Yu, 2014)

Classic ref: *(Bertsekas, Tsitsiklis, 1987)*

D-SGD:

- workers compute (stochastic) gradients
- server computes parameter update
- widely used (centralized) design choice
- can have quite high communication cost

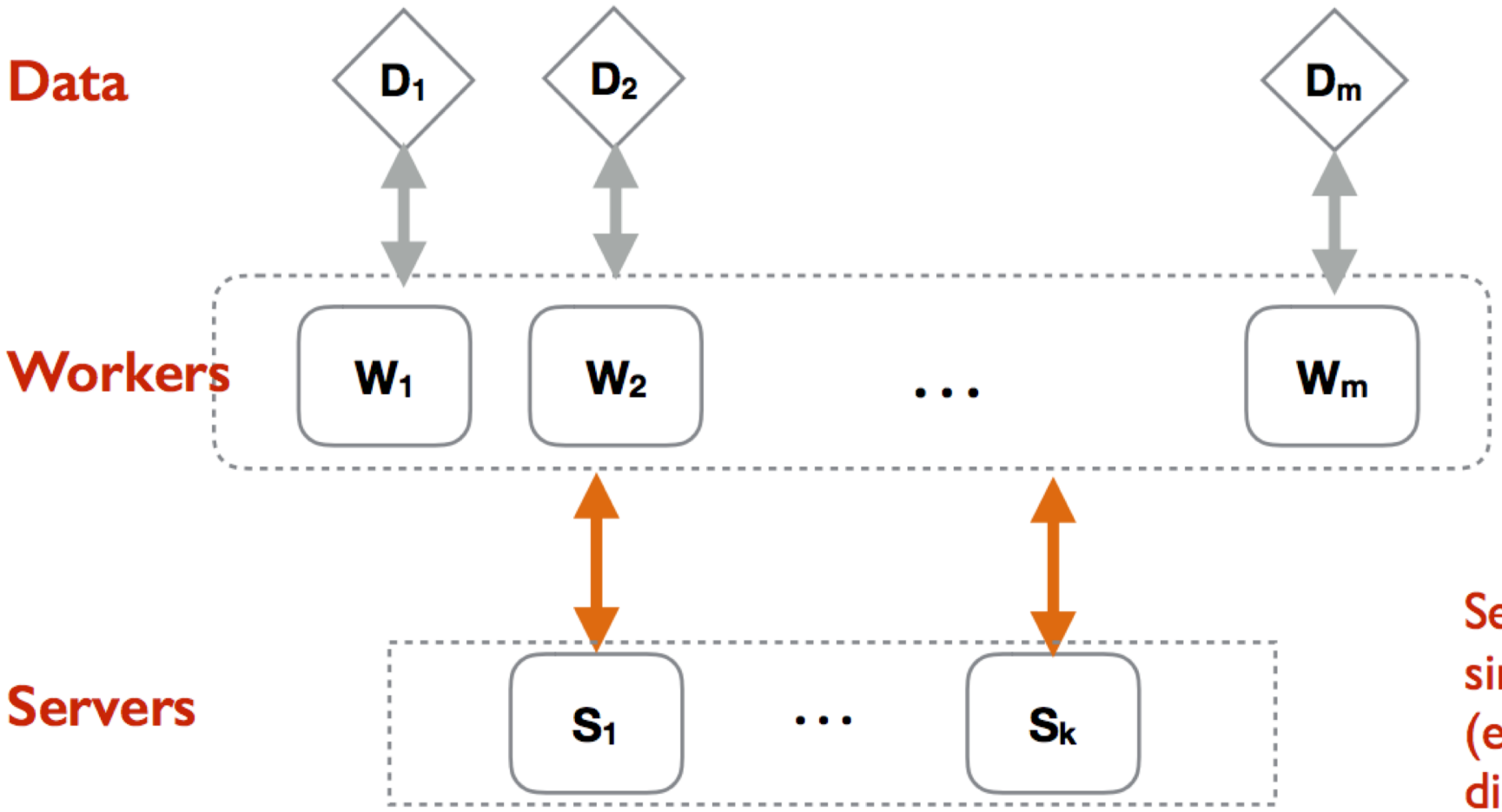
Asynchrony via: servers **use delayed / stale gradients** from workers

(Nedic, Bertsekas, Borkar, 2000; Agarwal, Duchi 2011) and many others

Slides Credit: Suvrit Sra (http://suvrit.de/talks/vr_nips16_sra.pdf)

Parallelization

To reduce communication, following idea is useful:



Worker nodes solve compute intensive subproblems

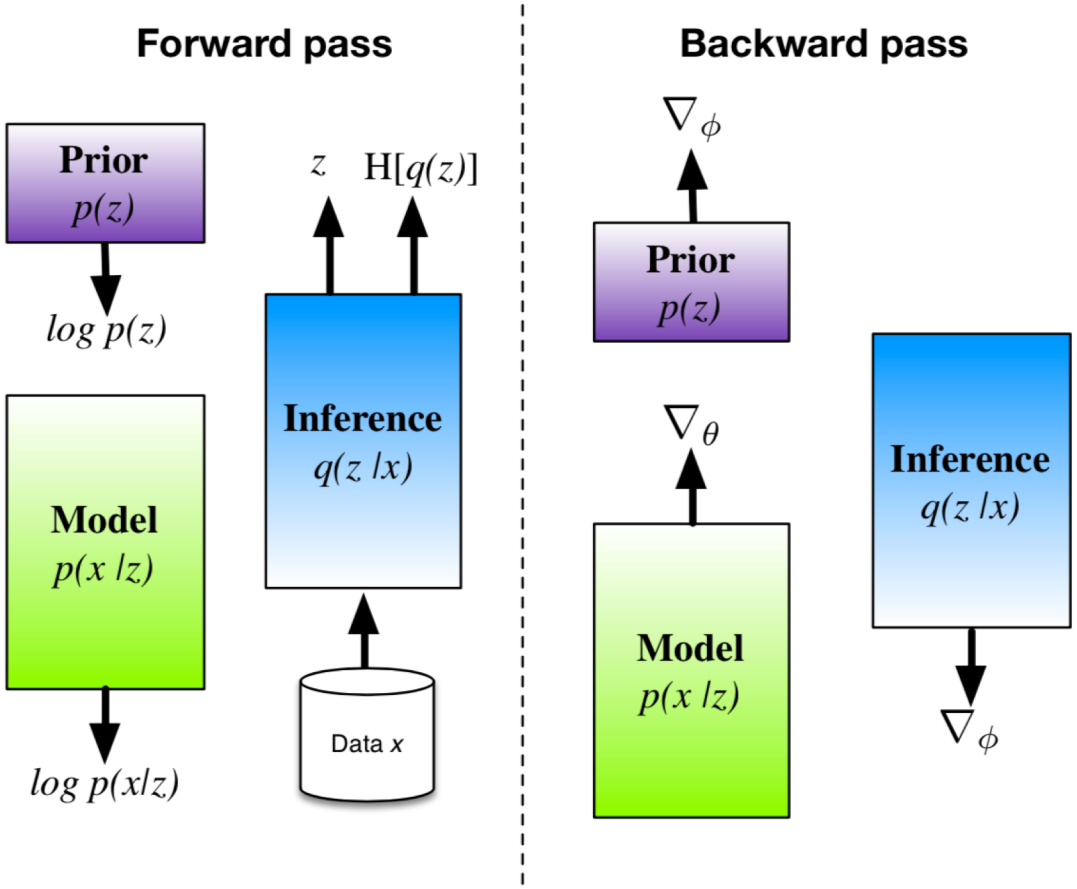
Servers perform simple aggregation (eg. full-gradients for distributed SVRG)

Slides Credit: Suvrit Sra (http://suvrit.de/talks/vr_nips16_sra.pdf)

Note on Implementation

$$\mathbb{E}_q[(-\log p(y|z) + \log q(z) - \log p(z)]$$

- Stochastic gradient descent and other preconditioned optimization.
- Probabilistic models are modular, can easily be combined.
- Same code can run on both GPUs or on distributed clusters.



Slides Credit: Shakir Mohamed (DeepMind)

Summary

Advantages and Disadvantages of VI

Disadvantages:

- An **approximate posterior** only - not always
- **Difficulty in optimisation** — can get stuck in guaranteed to find exact posterior in the limit. local minima.
- Typically **under-estimates the variance** of the posterior and can bias maximum likelihood parameter estimates.
- **Limited theory** and guarantees for variational methods.

Advantages:

- Applicable to almost **all probabilistic models**: non-linear, non-conjugate, high-dimensional, directed and undirected.
- Can be **faster to converge** than competing methods.
- Easy **convergence assessment**.
- **Numerically stable**.
- Can be used on **modern computing architectures** (CPUs and GPUs).
- Principled and scalable approach for **model selection**.

Mean field vs LBP

- LBP minimizes the **Bethe** energy while MF maximizes the **ELBO**.
- LBP is exact for trees whereas MF is not, suggesting LBP will in general.
- LBP optimizes over node and edge marginals, whereas naïve MF only optimizes over node marginals, again suggesting LBP will be more accurate.
- MF objective has many more local optima than the LBP objective, so optimizing the MF objective seems to be harder.
- MF tends to be more overconfident than BP
- The advantage of MF is that it gives a lower bound on the partition function while for LBP we don't know the relationship.
- MF is easier to extend to other distributions besides discrete and Gaussian.