

Chapter 19

Approximate Inference

Inference

- Inference refers to computing conditional distributions
- For example, in probabilistic models with latent \mathbf{h} and visible \mathbf{v} units, we usually wish to compute

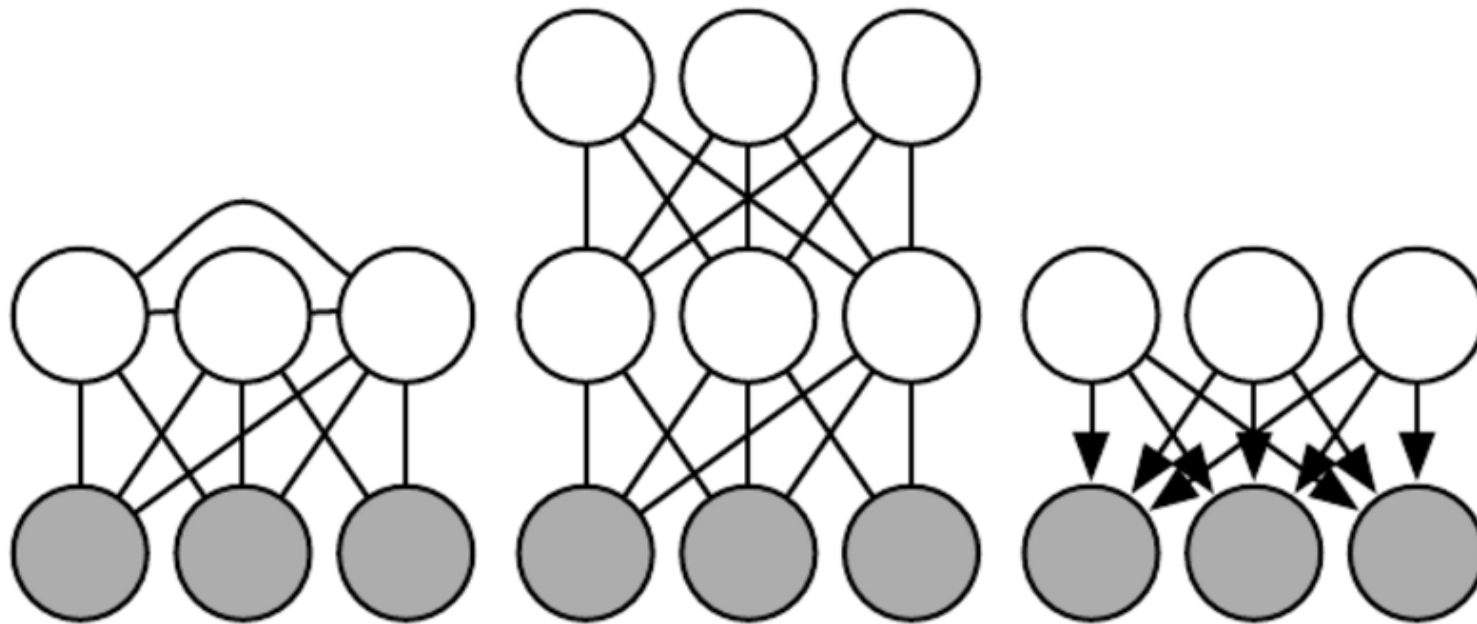
$$p(\mathbf{h}|\mathbf{v})$$

- An alternative form of inference is to compute

$$\mathbf{h}^* = \arg \max_{\mathbf{h}} p(\mathbf{h}|\mathbf{v})$$

- In graphical models with multiple layers of hidden units, it is usually difficult to evaluate them exactly

- The difficulty arises from the interactions between latent variables \mathbf{h} in computing the posterior $p(\mathbf{h}|\mathbf{v})$





Inference as Optimization

- Recall from the EM algorithm

$$\log p(\mathbf{v}; \boldsymbol{\theta}) = \mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta}) + \text{KL}(q(\mathbf{h}) || p(\mathbf{h}|\mathbf{v}; \boldsymbol{\theta}))$$

where

$$\begin{aligned}\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta}) &= \int q(\mathbf{h}) \log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) d\mathbf{h} - \int q(\mathbf{h}) \log q(\mathbf{h}) d\mathbf{h} \\ &= E_{\mathbf{h} \sim q} \log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) + H(q)\end{aligned}$$

$$\text{KL}(q(\mathbf{h}) || p(\mathbf{h}|\mathbf{v}; \boldsymbol{\theta})) = \int q(\mathbf{h}) \log \frac{q(\mathbf{h})}{p(\mathbf{h}|\mathbf{v}; \boldsymbol{\theta})} d\mathbf{h}$$

- Some changes of variables (e.g. $\mathbf{X} \rightarrow \mathbf{v}$ and $\mathbf{Z} \rightarrow \mathbf{h}$) have been made for consistent notation

- Since the KL divergence is non-negative, $\text{KL}(q||p) \geq 0$, it follows that

$$\log p(\mathbf{v}; \boldsymbol{\theta}) \geq \mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta})$$

with equality if and only if

$$q(\mathbf{h}) = p(\mathbf{h}|\mathbf{v}; \boldsymbol{\theta})$$

- In other words, $\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta})$ is a lower bound on $\log p(\mathbf{v}; \boldsymbol{\theta})$ for **any choice of $q(\mathbf{h})$**
- The bound \mathcal{L} is closer to $\log p(\mathbf{v}; \boldsymbol{\theta})$ when $q(\mathbf{h})$ is closer to $p(\mathbf{h}|\mathbf{v}; \boldsymbol{\theta})$

- Inference
 - Exact inference maximizes $\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta})$ perfectly by searching over a family of functions $q(\mathbf{h})$ that includes $p(\mathbf{h}|\mathbf{v}; \boldsymbol{\theta})$
 - Approximate inference makes optimization approximate by restricting the $q(\mathbf{h})$'s to search over, or by imperfect optimization
- Learning
 - Maximize $\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ for a given $q(\mathbf{h})$
- Combined inference and learning is possible via algorithms such as coordinate ascent

Revisiting Kullback-Leibler (KL) Divergence

- In maximizing $\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta})$, we are minimizing $\text{KL}(q||p)$, which yields an effect different from minimizing $\text{KL}(p||q)$

$$q^* = \arg \min_q \text{KL}(p||q)$$

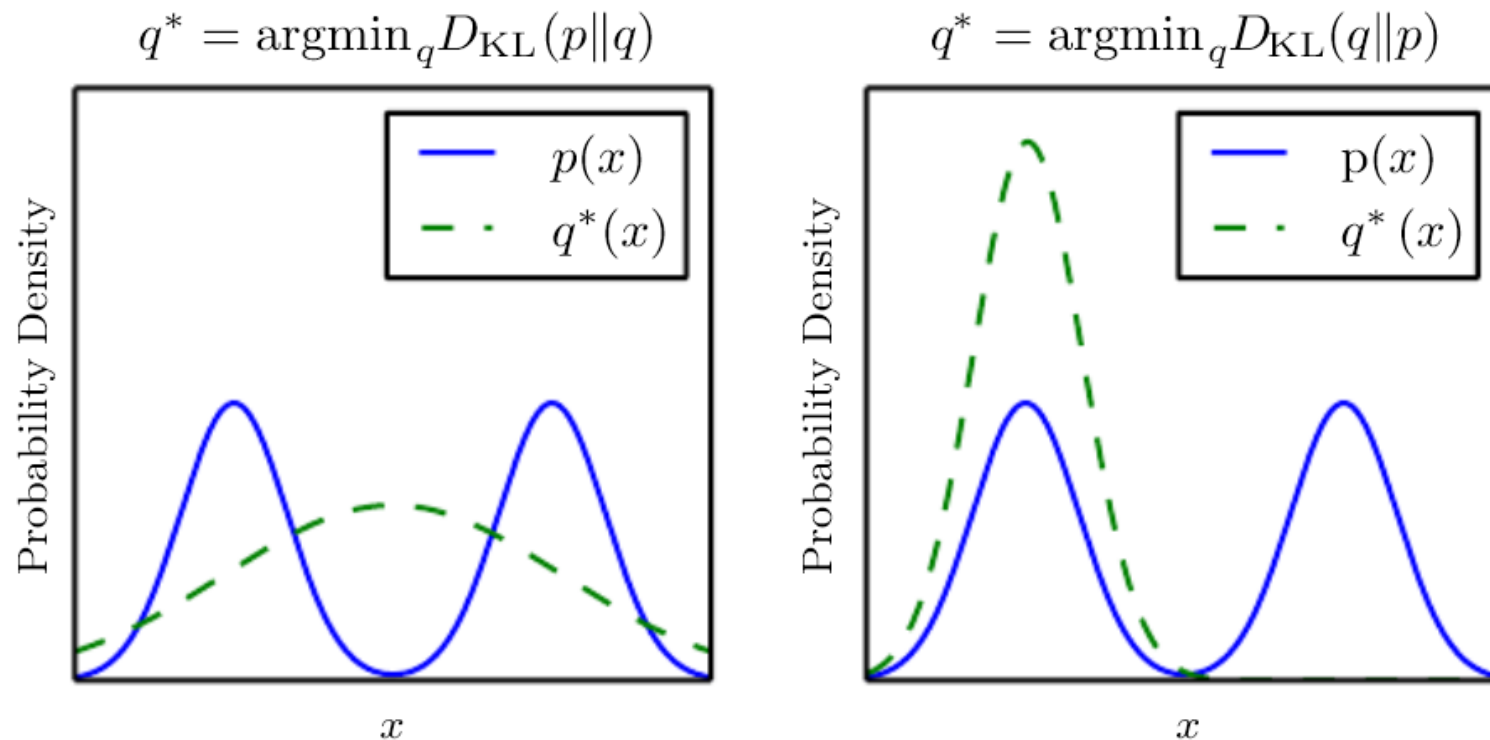
v.s.

$$q^* = \arg \min_q \text{KL}(q||p)$$

where

$$\text{KL}(p||q) = \int p(x) \left(\log \frac{1}{q(x)} - \log \frac{1}{p(x)} \right) dx$$

$$\text{KL}(q||p) = \int q(x) \left(\log \frac{1}{p(x)} - \log \frac{1}{q(x)} \right) dx$$



- $\text{KL}(p||q)$ places high probability where data occur
- $\text{KL}(q||p)$ places low probability where data do not occur

Calculus of Variations

- Learning algorithms are mainly based on minimizing a function $J(\theta)$ by solving for the critical points

$$\nabla_{\theta} J(\theta) = 0$$

- Calculus of variations solves for a function $f(x)$ to maximize a functional $J(f(x))$ by taking functional derivatives, a.k.a **variational derivatives**, w.r.t. the value of the function f at point x

$$\frac{\delta}{\delta f(x)} J$$

- Example: To find a $p(x)$ that maximizes differential entropy

$$\arg \max_{p(x)} H[p(x)] = - \int p(x) \log p(x) dx$$

subject to

$$\int p(x) dx = 1$$

$$E[x] = \mu$$

$$E[(x - \mu)^2] = \sigma^2$$

- This constrained optimization problem can be solved by setting up a Lagrangian functional

$$\begin{aligned} \mathcal{L}(p, \lambda_1, \lambda_2, \lambda_3) = & - \int p(x) \log p(x) dx + \lambda_1 \left(\int p(x) dx - 1 \right) \\ & + \lambda_2 \left(\int x p(x) dx - \mu \right) + \lambda_3 \left(\int (x - \mu)^2 p(x) dx - \sigma^2 \right) \end{aligned}$$

- We then set the functional derivative w.r.t. $p(x)$ to 0

$$\forall x, \frac{\delta}{\delta p(x)} \mathcal{L} = -\log p(x) - 1 + \lambda_1 + \lambda_2 + \lambda_3(x - \mu)^2 = 0$$

- It is seen that the optimal $p(x)$ has a form of Gaussian and can be solved exactly as a Gaussian after all the λ 's are solved

$$p(x) = \exp(\lambda_1 + \lambda_2 x + \lambda_3(x - \mu)^2 - 1) = \mathcal{N}(x; \mu, \sigma^2)$$

Variational Inference

- Variational inference is to find $p(\mathbf{h}|\mathbf{v})$ by maximizing $\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta})$ w.r.t. $q(\mathbf{h})$, which can be any function of \mathbf{h}

$$\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta}) = E_{\mathbf{h} \sim q} \log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) - H(q)$$

- An approximate framework (known as the **mean field approximation**) of variational inference assumes $q(\mathbf{h})$ is factorial

$$q(\mathbf{h}) = \prod_{i=1}^M q_i(\mathbf{h}_i)$$

where the elements of \mathbf{h} is partitioned into disjoint groups \mathbf{h}_i 's

- **No further assumption is made about the forms of $q_i(\mathbf{h}_i)$**



- We now seek the $q_i(\mathbf{h}_i)$ for which the lower bound is the largest by making a free form (variational) optimization w.r.t. each of them

$$\begin{aligned}
 & \mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta}) \\
 &= \int q(\mathbf{h}) \log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) d\mathbf{h} - \int q(\mathbf{h}) \log q(\mathbf{h}) d\mathbf{h} \\
 &= \int \prod_i q_i(\mathbf{h}_i) \left\{ \log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) - \sum_i \log q_i(\mathbf{h}_i) \right\} d\mathbf{h} \\
 &= \int q_j(\mathbf{h}_j) \left\{ \int \log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) \prod_{i \neq j} q_i(\mathbf{h}_i) d\mathbf{h}_i \right\} d\mathbf{h}_j \\
 &\quad - \int q_j(\mathbf{h}_j) \log q_j(\mathbf{h}_j) d\mathbf{h}_j + \text{const} \\
 &= \int q_j(\mathbf{h}_j) \log \tilde{p}(\mathbf{v}, \mathbf{h}_j) d\mathbf{h}_j - \int q_j(\mathbf{h}_j) \log q_j(\mathbf{h}_j) d\mathbf{h}_j + \text{const} \\
 &= -\text{KL}(q_j(\mathbf{h}_j) \parallel \tilde{p}(\mathbf{v}, \mathbf{h}_j)) + \text{const}
 \end{aligned}$$

where we have defined

$$\begin{aligned}\log \tilde{p}(\mathbf{v}, \mathbf{h}_j) &= \int \log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) \prod_{i \neq j} q_i(\mathbf{h}_i) d\mathbf{h}_i \\ &= E_{i \neq j}(\log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta})) + \text{const}\end{aligned}$$

- It is seen that \mathcal{L} is maximized when

$$q_j(\mathbf{h}_j) = \tilde{p}(\mathbf{v}, \mathbf{h}_j) = \frac{1}{Z} \exp(E_{i \neq j}(\log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta})))$$

- Alternatively, we have the unnormalized distribution $\tilde{q}_j(\mathbf{h}_j)$

$$\tilde{q}_j(\mathbf{h}_j) = \exp(E_{i \neq j}(\log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta})))$$

- Carrying out the expectation $E_{i \neq j}(\log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}))$ will yield the correct functional form for $\tilde{q}_j(\mathbf{h}_j)$

- Toy problem: Consider a probabilistic model with latent variables $\mathbf{h} \in R^2$ and one visible variable $v \in R^1$

$$p(\mathbf{h}) = \mathcal{N}(\mathbf{h}; \mathbf{0}, \mathbf{I})$$

$$p(v|\mathbf{h}) = \mathcal{N}(v; \mathbf{w}^T \mathbf{h}, 1)$$

- The true posterior is given by

$$\begin{aligned} p(\mathbf{h}|v) &= \frac{p(\mathbf{h})p(v|\mathbf{h})}{p(v)} \\ &= \mathcal{N}(\mathbf{h}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \end{aligned}$$

where

$$\boldsymbol{\Sigma}^{-1} = \mathbf{I} + \mathbf{w}\mathbf{w}^T$$

$$\boldsymbol{\mu} = v\boldsymbol{\Sigma}\mathbf{w}$$

- Obviously, $p(\mathbf{h}|v)$ is not factorial



- Now, applying variational inference, we have

$$p(\mathbf{h}|v) \approx q(\mathbf{h}|v) = q(h_1|v)q(h_2|v)$$

$$\begin{aligned}\tilde{q}(h_1|v) &= \exp(E_{h_2 \sim q(h_2|v)}(\log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}))) \\ &= \exp\left(-\frac{1}{2}E_{h_2 \sim q(h_2|v)}[h_1^2 + h_2^2 + v^2 + h_1^2 w_1^2 + h_2^2 w_2^2 \right. \\ &\quad \left. - 2vh_1 w_1 - 2vh_2 w_2 + 2h_1 w_1 h_2 w_2]\right) \\ &= \exp\left(-\frac{1}{2}[h_1^2 + \langle h_2^2 \rangle + v^2 + h_1^2 w_1^2 + \langle h_2^2 \rangle w_2^2 \right. \\ &\quad \left. - 2vh_1 w_1 - 2v\langle h_2 \rangle w_2 + 2h_1 w_1 \langle h_2 \rangle w_2]\right) \\ &= \exp\left(-\frac{1}{2}[(1 + w_1^2)h_1^2 - 2(vw_1 - w_1 \langle h_2 \rangle w_2)h_1 + \text{const}]\right)\end{aligned}$$

- From the above, we can deduce

$$q(h_1|v) = \mathcal{N}(h_1; \mu_1, \sigma_1^2)$$



where

$$\sigma_1^2 = \frac{1}{1 + w_1^2}$$

$$\mu_1 = \frac{vw_1 - w_1 \langle h_2 \rangle w_2}{1 + w_1^2} = \frac{vw_1 - w_1 \mu_2 w_2}{1 + w_1^2}$$

- On the symmetry ground, we can readily arrive at

$$q(h_2|v) = \mathcal{N}(h_2; \mu_2, \sigma_2^2)$$

where

$$\sigma_2^2 = \frac{1}{1 + w_2^2}$$

$$\mu_2 = \frac{vw_2 - w_2 \langle h_1 \rangle w_1}{1 + w_2^2} = \frac{vw_2 - w_2 \mu_1 w_1}{1 + w_2^2}$$

- Note that we do not assume $q(\mathbf{h}|v) = q(h_1|v)q(h_2|v)$ is Gaussian; its Gaussian form is derived automatically



- Moreover, given the data v and the model parameters w_1, w_2 , we have the following fixed-point update equations to estimate u_1, u_2 iteratively

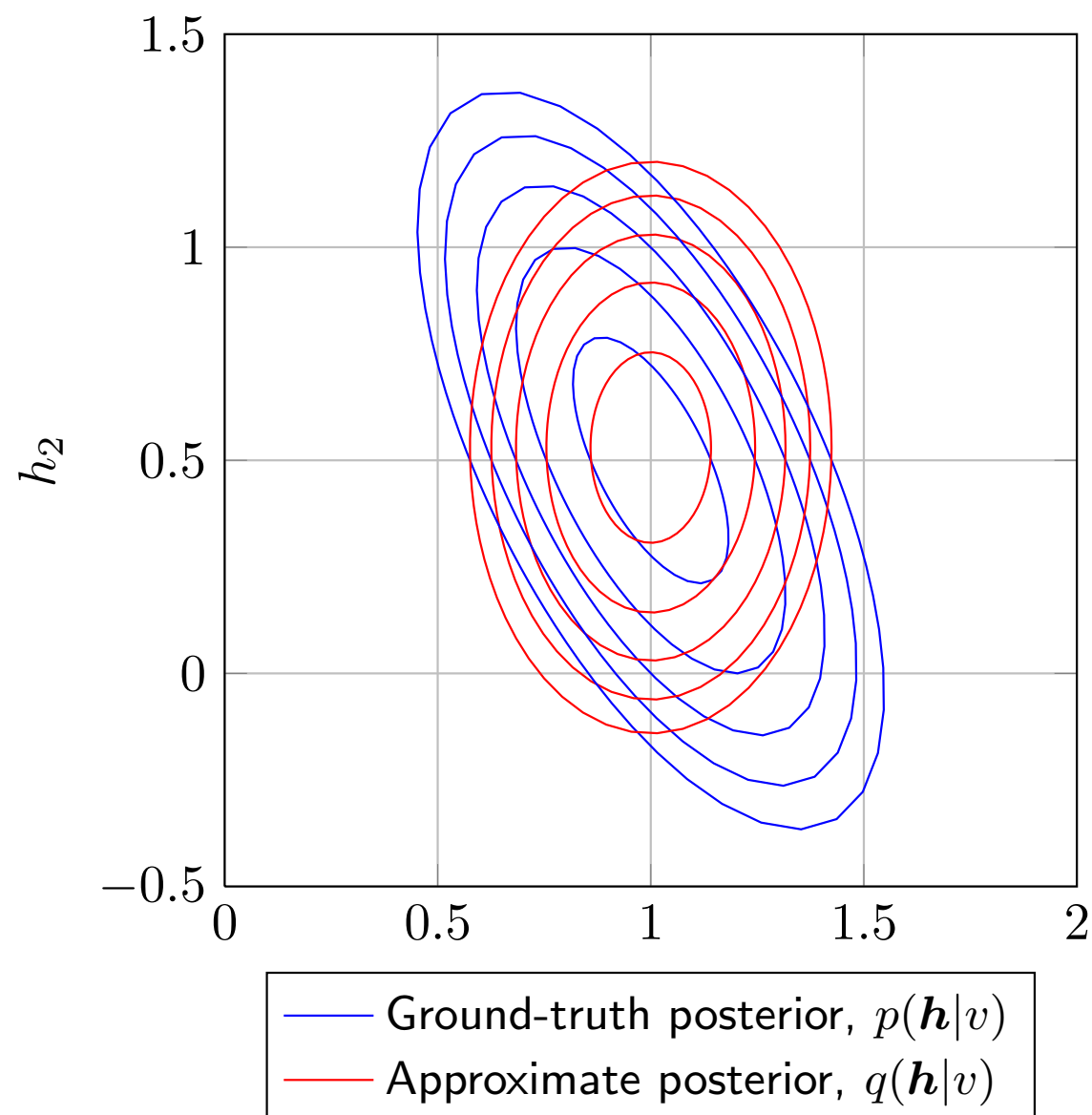
$$\mu_1 = \frac{vw_1 - w_1\mu_2w_2}{1 + w_1^2}$$

$$\mu_2 = \frac{vw_2 - w_2\mu_1w_1}{1 + w_2^2}$$

- Note that the process needs to be carried out for each new value of v
- The following compares the ground-true posterior $p(\mathbf{h}|v)$ with the variational approximation $q(\mathbf{h}|v)$, with

$$\mathbf{w} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad v = 3$$

and the fixed-point update equations run for 10 iterations



Learned Approximate Inference

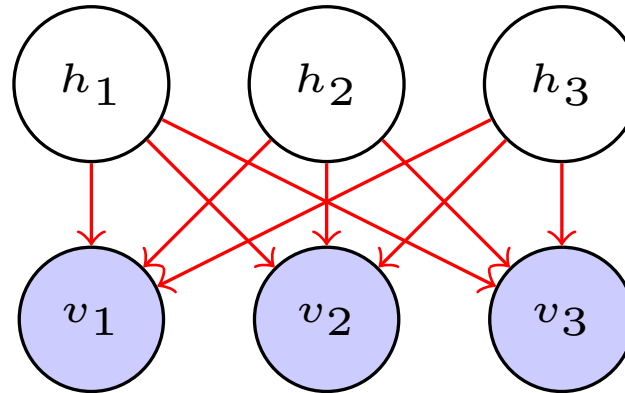
- **Idea:** To replace the iterative optimization with a neural network $f(\mathbf{v}; \boldsymbol{\theta}')$ to form an approximate inference distribution $q(\mathbf{h}|\mathbf{v})$
- **Wake-Sleep algorithm** (in sleep mode) is to learn the mapping from \mathbf{v} to \mathbf{h} by drawing \mathbf{v}, \mathbf{h} samples from the model distribution $p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta})$ while the model is still evolving in the learning process
- One drawback is that we will only be able to train $f(\mathbf{v}; \boldsymbol{\theta}')$ on \mathbf{v} 's that have high probability under the evolving model $p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta})$

Sparse Coding and MAP Inference

- Sparse coding can be interpreted as a special case of **combined variational inference and learning**
- Combined variational inference and learning
 1. Find approximate inference q by maximizing $\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta})$ w.r.t. q
 2. Find model parameters $\boldsymbol{\theta}$ by maximizing $\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$
 3. Repeat steps 1, 2 until the stopping criteria are met



- Linear factor model for sparse coding



$$p(\mathbf{h}) = \prod_i \frac{\lambda}{2} e^{-\lambda|h_i|}$$

$$p(\mathbf{v}|\mathbf{h}) = \mathcal{N}(\mathbf{v}; \mathbf{W}\mathbf{h} + b, \beta^{-1}\mathbf{I})$$

where λ, β are hyper parameters and are assumed to be known

- In learning the model parameters \mathbf{W}, b , the EM does not work because $p(\mathbf{h}|\mathbf{v})$ is intractable

- We thus resort to approximate variational inference by restricting $q(\mathbf{h}) \approx p(\mathbf{h}|\mathbf{v})$ to the following family

$$q(\mathbf{h}; \boldsymbol{\mu}) = \delta(\mathbf{h} - \boldsymbol{\mu})$$

where $\delta(\cdot)$ is the Dirac delta function

- Maximizing $\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta}) = E_{\mathbf{h} \sim q} \log p(\mathbf{v}, \mathbf{h}; \boldsymbol{\theta}) + H(q)$ w.r.t. q arrives at

$$\mathbf{u}^* = \arg \max_{\boldsymbol{\mu}} \log p(\mathbf{v}, \mathbf{h} = \mathbf{u}; \boldsymbol{\theta}) = \underbrace{\arg \max_{\boldsymbol{\mu}} p(\mathbf{h} = \mathbf{u}|\mathbf{v}; \boldsymbol{\theta})}_{\text{MAP inference}}$$

$$= \arg \max_{\boldsymbol{\mu}} -\lambda \sum_i |u_i| - \frac{\beta}{2} \|\mathbf{v} - \mathbf{W}\mathbf{u} - \mathbf{b}\|_2^2$$

$$= \arg \min_{\boldsymbol{\mu}} \lambda \sum_i |u_i| + \frac{\beta}{2} \|\mathbf{v} - \mathbf{W}\mathbf{u} - \mathbf{b}\|_2^2$$

- With this $q(\mathbf{h}; \boldsymbol{\mu}^*) = \delta(\mathbf{h} - \boldsymbol{\mu}^*)$, maximizing $\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ gives

$$\begin{aligned} & \arg \max_{\boldsymbol{\theta}} \log p(\mathbf{v}, \mathbf{h} = \boldsymbol{\mu}^*; \boldsymbol{\theta}) \\ &= \arg \min_{\mathbf{W}, \mathbf{b}} \lambda \sum_i |u_i^*| + \frac{\beta}{2} \|\mathbf{v} - \mathbf{W} \mathbf{u}^* - \mathbf{b}\|_2^2 \end{aligned}$$

- To summarize, we alternate between minimization w.r.t. \mathbf{u} and \mathbf{W}, \mathbf{b}

$$J(\mathbf{u}, \mathbf{W}, \mathbf{b}) = \lambda \sum_i |u_i| + \frac{\beta}{2} \|\mathbf{v} - \mathbf{W} \mathbf{u} - \mathbf{b}\|_2^2$$

where \mathbf{u} is known as the sparse code for the input \mathbf{v}

- The procedure corresponds to the conventional sparse coding and a form of coordinate ascent on $\mathcal{L}(\mathbf{v}, q, \boldsymbol{\theta})$

Interaction between Learning and Inference

- Using approximate inference as part of the learning algorithm affects the learning process; this in turn affects the inference accuracy
- In learning the model parameters, variational learning increases

$$E_{\mathbf{h} \sim q(\mathbf{h})} \log p(\mathbf{v}, \mathbf{h})$$

- For a specific \mathbf{v} , this has an effect of learning a model that
 - Increases $p(\mathbf{h}|\mathbf{v})$ where $q(\mathbf{h}|\mathbf{v})$ has high probability
 - Decreases $p(\mathbf{h}|\mathbf{v})$ where $q(\mathbf{h}|\mathbf{v})$ has low probability
- As a result, the approximate assumptions (e.g. the mean field assumption) become self-fulfilling prophecies
- The true amount of harm due to approximate inference is generally hard to estimate

Review

- Inference
- Inference as optimization
- $\text{KL}(q||p)$ vs $\text{KL}(p||q)$
- Calculus of variations
- Variational inference
- Combined variational inference and learning