Variational Inference:

Aim: We want to know $P(Z|X)$.

Given: Prior of $P(Z)$: A prior knowledge of what $P(Z)$ is like, no business with observed data X. A candidate searching family $q_\theta(Z)$, which means that what kind of distribution is most likely to be used to approximate $P(Z|X)$.

Theory dependence:

- Measuring how $q(Z)$ approximate the true target $P(Z|X)$ → kL divergence.

$$KL(q(Z)\,||\,P(Z|X)) = E_q(\log q(Z)) - E_q(\log P(Z|X))$$

we want this to be replaced.

$$= E_q(\log q(Z)) - E_q(\log P(Z,X)) + \log P(X)$$

$\log P(X)$ || Intractable

Then we get concept of ELBO:

$$ELBO(q(Z)) = E(\log P(X,Z)) - E(\log q(Z))$$
$$= E(\log P(X|Z)) + E(\log P(Z)) - E(\log q(Z))$$
$$= E(\log P(X|Z)) + KL(q\,||\,P(Z))$$

- From above, we can see that we continuously searching optimal q_θ. monitoring ELBO (indicating how near it is between q and p) to decide when we stop updating q with parameters.

Algorithms:

Initialize parameters deciding $q(Z|\theta) \Rightarrow \theta_0$.

mean filed assumption

$$q(Z) = \prod_{i=1}^{m} q(Z_i)$$

Then we get full joint prob

WHILE NOT converge:

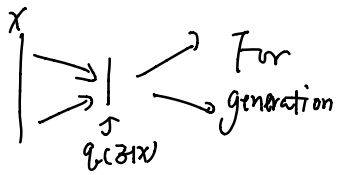
updating $q(Z|\theta)$ using updating rules for θ

computing ELBO to decide whether it is converged.

← where the prior of $P(Z)$ engaged.

Variational Autoencoders:

Proposal of an ABUS algorithm which can optimize learning:



Inference as optimization

Here inference is proposed as an optimation problem, which means that we want to find optimal q, which can minimize $KL(q||P)$

Expectation maximization

EM can be seen as combination of two steps:

E step: Inference $q(h|v) = p(h|v)$ (Here in EM we can particularly calculate the exact form of $p(h|v)$, thus inference of $p(h|v)$ directly done, which also means that $KL(q||p)=0$ in E step.

M step: Learning algorithms using maximum likelihood, $\theta^* = \arg\max_\theta p(v|\theta,h)$, after updating $\theta=\theta^*$. $q(h|v) \neq p(h|v,\theta^*)$, thus $KL(q||p)\neq 0$, again back to E step to do the inference.

MAP inference and Sparse coding

- MAP inference: Instead of speculating full distribution $p(h|v)$, we just use one of the maximum to stand for $h^* = \arg\max_h p(h|v)$
- Sparse Coding:

  Sparse Priority: $p(h_i) = \frac{\lambda}{2} e^{-\lambda|h_i|}$

  Data distribution: $p(x|h) = \mathcal{N}(v|Wh+b, \beta^{-1}I)$

  Sparse coding minimize:

  $J(H,W) = \sum_{i,j}|H_{i,j}| + \sum_{i,j}(V-HW^T)^2_{i,j}$ Optimizing this problem by iteratively maximizing wrt H and wrt W.

Variational Inference and Learning

- Term clarification: Inference, we always refer to find distribution $p(h|v)$, learning we refer to find optimal parameters θ.
- Two KL divergence stand for:

$KL(q\|p)$: q must be small where p is small (The divergence we need to minimize when we do variational inference)

$KL(p\|q)$: q must be large where p is large (The divergence we want to minimize when we are learning with maximum likelihood)

- Discrete Latent Variables
  - In simplest case, h_i is binary indicating which mode is engaged, we can assume $q(h_i=1|v) = \hat{h}_i$, then solving optimization problems with iteratively fixed point: $\frac{\partial}{\partial \hat{h}_i} L = 0$
  - Binary sparse coding as an example:

    $p(h_i=1) = \delta(b_i)$ → parameters for learning

    $p(v|h) = \mathcal{N}(v|Wh, \beta^{-1})$ Observed data consists of summation of different Gaussians

    Then for maximizing likelihood:

$$\frac{\partial}{\partial b_i}\log p(v) = \frac{\frac{\partial}{\partial b_i}p(v)}{p(v)} = \frac{\frac{\partial}{\partial b_i}\sum_h p(v|h)\cdot p(h)}{p(v)}$$

$$= \frac{\sum_h p(v|h)\frac{\partial}{\partial b_i}p(h)}{p(v)} = \mathbb{E}_{h\sim p(h|v)}\frac{\partial}{\partial b_i}\log p(h)$$

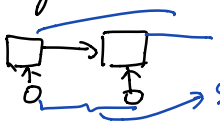
Here we need to inference $p(h|v)$

⇒ When engaging mean-field approximation, and we select $q(h_i=1|v) = \hat{h}_i$, we get: $\mathcal{L}(v, \theta, q) = \mathbb{E}(\log p(h) + \log p(v|h) - \log q(h|v))$ for variational inference.

prior — likelihood — estimation

updating q, then learning b_i, we get the final solution.

- Relationship with RNN: According to the updating $q(h_i|v)$ rule, we can see that we update h_i iteratively, which means that we change $q(h|v)$ one "h" per time according to last time results:



- Relationship with autoencoder: $x \rightarrow$ encodes by hi modified $\rightarrow$ decode for optimizing $\theta \rightarrow$ encode for hj modified $\rightarrow \cdots\cdots \rightarrow$ converge

- Calculus of variations

  A math technique used for continuous variational inference when updating q.

- Continuous latent variables

  Same algorithms, use calculus of variations to maximize $\mathcal{L}$ with respect to q.

- Interaction between learning and inferencing.

  When we are learning parameters θ^*, it is hard to judge if our variational inference works well, if the true optimal θ^* is too complicated for our q to learn $p(h|v,\theta^*)$ our learning will fail or mis-leading.

Learned Approximate Inference

- What neural network can help us?

  When using variational inference, it's like a mapping from all v to $q^*(h|v,\theta)$. namely the calculation process of q when given v (or searching process), this mapping $q^* = f(v,\theta)$ can be modeled using neural network.

- Wake-sleep

  Here comparing biological dreaming to calculating $p(v,h)$ and inference $p(h|v)$, awaking to be learning θ, which maximizes $p(v|h,\theta)$

- VAE: Definition of loss $p(z|x)$ and $q(z|x)$, then use neural networks to learn it by itself.