variational bayesian

Variational Bayesian methods - Wikipedia

Variational Bayesian methods are a family of techniques for approximating intractable integrals arising in Bayesian inference and machine learning.

In the former purpose (that of approximating a posterior probability), variational Bayes is an alternative to Monte Carlo sampling methods — particularly, Markov chain Monte Carlo methods such as Gibbs sampling — for taking a fully Bayesian approach to statistical inference over complex distributions that are difficult to evaluate directly or sample. In particular, whereas Monte Carlo techniques provide a numerical approximation to the exact posterior using a set of samples, Variational Bayes provides a locally-optimal, exact analytical solution to an approximation of the posterior.

(和蒙特卡龙方法进行对比)

根据 变分：Variational Bayesian methods - Wikipedia

$$q_{\boldsymbol{v}_i}^*(\boldsymbol{v}_i) = \frac{\exp(\int_{-\boldsymbol{v}_i} q(\boldsymbol{v}) q(\boldsymbol{v}, \boldsymbol{y}))}{\int_{\boldsymbol{v}_i} \exp\left(\int_{-\boldsymbol{v}_i} q(\boldsymbol{v}) q(\boldsymbol{v}, \boldsymbol{y})\right)} \propto \exp\left(\int_{-\boldsymbol{v}_i} q(\boldsymbol{v}) q(\boldsymbol{v}, \boldsymbol{y})\right)$$

隐变量先验概率为：

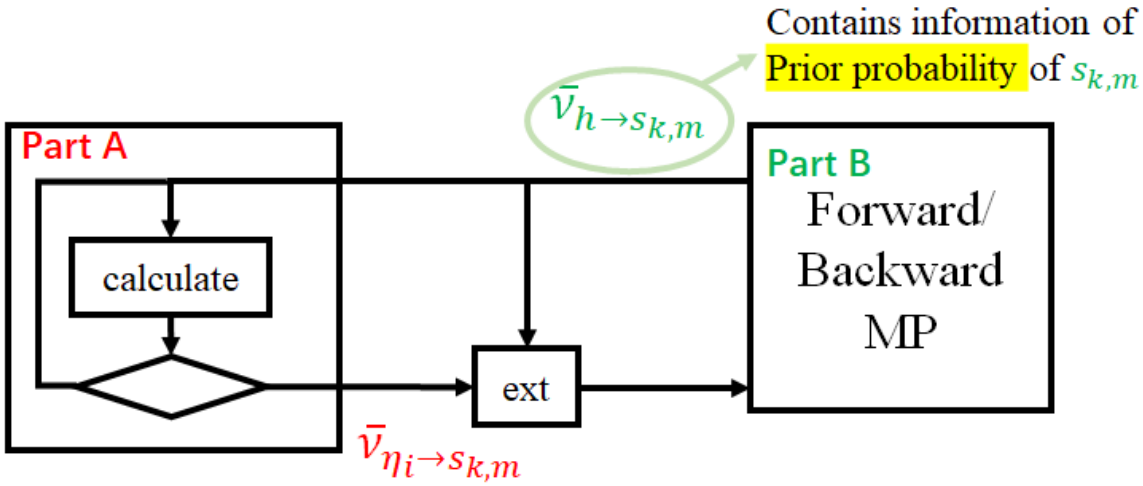
$$\hat{p}(\boldsymbol{v}) = p(\boldsymbol{x} \mid \boldsymbol{\gamma}) p(\boldsymbol{\kappa}) p(\boldsymbol{\gamma} \mid \boldsymbol{s}) p(\boldsymbol{c}, \boldsymbol{s}; \boldsymbol{\xi})$$

隐变量后验概率，通过Mean Field approximation 分解为：

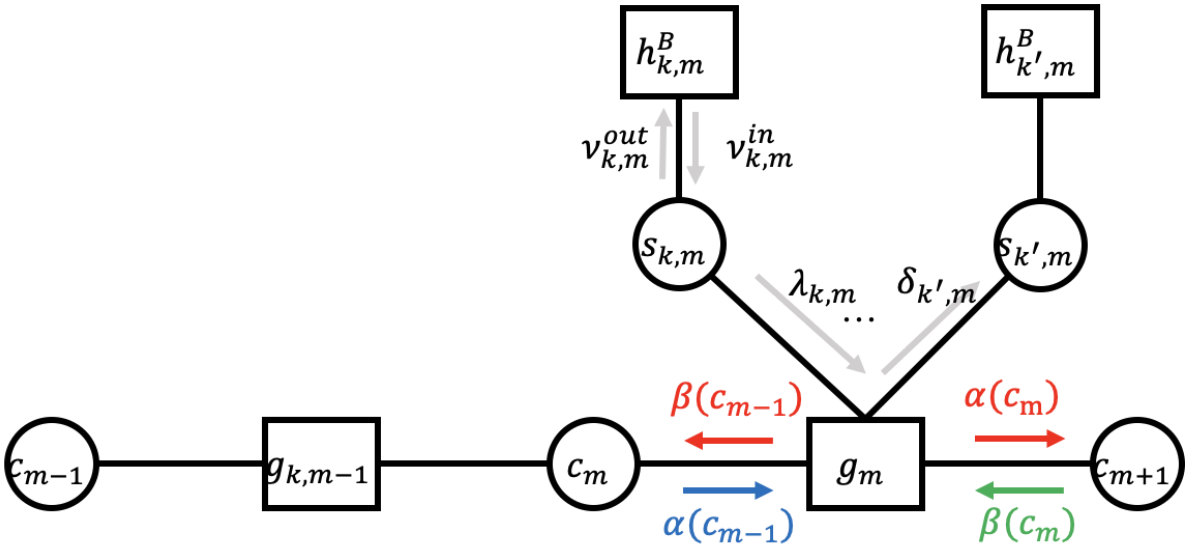
$$q(\boldsymbol{v}) = q(\boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{c}, \boldsymbol{s}) = q(\boldsymbol{x}) q(\boldsymbol{\gamma}) q(\boldsymbol{c}) q(\boldsymbol{s})$$

注意不要搞混了。

Model A and Model B



Module A



为了加快计算和绕过带有loop的factor graph，我们将整个factor graph 分为两部分，其中：

$$h^A_{k,m}(s_{k,m}) \triangleq \nu^{out}_{k,m}(s_{k,m})$$
$$h^B_{k,m}(s_{k,m}) \triangleq \nu_{\eta_{k,m}\to s_{k,m}}(s_{k,m})$$

Model B 中传递给module A的消息可以使用sum-product-rule [1] 来计算：

为了更高效地利用forward/backward 方法 [1] 计算HMM中的相关性，我们利用一个综合的factor node g_m 来刻画 $c_m, c_{m+1}, s_{k,m}$ 的相关性：

$$\begin{aligned} g_m(c_{m+1}, c_m, s_{1,m}, \ldots, s_{K,m}) &\triangleq p(c_{m+1} \mid c_m) p(s_{1,m}, \ldots, s_{K,m} \mid c_m) p(c_m) \\ &= p(c_{m+1} \mid c_m) p(c_m) \prod_k^K p(s_{k,m} \mid c_m) \end{aligned}$$

图中 $\nu^{in}_{k,m}(s_{k,m})$ 的值相等于 model A到 model B 的消息 $h^B_{k,m}(s_{k,m}) \triangleq \nu_{\eta_{k,m}\to s_{k,m}}(s_{k,m})$，然而 model A 为了提高算法效率，并避开带有loop的factor graph 求解，使用了 VBI，那么module A 处只能计算出 $s_{k,m}$ 的后验概率 $q(s_{k,m})$ 而不能直接得到 $\nu_{\eta_{k,m}\to s_{k,m}}(s_{k,m})$， 根据message 的定义：

$$\nu_{\eta_{k,m}\to s_{k,m}}(s_{k,m}) \cdot \nu_{h\to s_{k,m}}(s_{k,m}) = p^m(s_{k,m}) \propto q(s_{k,m})$$
$$\Longrightarrow \nu^{in}_{k,m}(s_{k,m}) = \nu_{\eta_{k,m}\to s_{k,m}}(s_{k,m}) \propto \frac{q(s_{k,m})}{\nu_{h\to s_{k,m}}(s_{k,m})}$$

其中，$\nu_{h\to s_{k,m}}$ 是上一次运算循环中 model B 传递给 model A的消息。

为了保证算法能够正常运行，我们需要将消息赋予一个确定的值，而不是一个正比关系，那么我们定义归一化的消息 $\bar{\nu}_{k,m}^{in}(s_{k,m})$:

$$\bar{\nu}_{k,m}^{in}(s_{k,m})=\frac{q(s_{k,m})\cdot\nu_{h\to s_{k,m}}(s_{k,m})}{\sum_{s_{k,m}}q(s_{k,m})\cdot\nu_{h\to s_{k,m}}(s_{k,m})}=\frac{q(s_{k,m})\cdot\bar{\nu}_{h\to s_{k,m}}(s_{k,m})}{\sum_{s_{k,m}}q(s_{k,m})\cdot\bar{\nu}_{h\to s_{k,m}}(s_{k,m})}$$

接下来定义前向消息（Farward）$\alpha(c_m)$:

$$\begin{aligned}\alpha\left(c_m\right)&=\sum_{-c_m}\left\{g_m\left(c_{m+1},c_m,s_{1,m},\ldots,s_{K,m}\right)\cdot\alpha\left(c_{m-1}\right)\cdot\prod_k\lambda_{k,m}\left(s_{k,m}\right)\right\}\\&\propto\sum_{-c_m}\left\{p(c_{m+1}\mid c_m)p(c_m)\prod_k^K p(s_{k,m}\mid c_m)\cdot\alpha(c_{m-1})\cdot\prod_k\bar{\nu}_{k,m}^{in}(s_{k,m})\right\}\end{aligned}$$

接下来定义后向消息（Backward） $\beta(c_{m-1})$:

$$\begin{aligned}\beta\left(c_{m-1}\right)&=\sum_{-c_{m-1}}\left\{g_m\left(c_{m+1},c_m,s_{1,m},\ldots,s_{K,m}\right)\cdot\beta\left(c_m\right)\cdot\prod_k\lambda_{k,m}\left(s_{k,m}\right)\right\}\\&\propto\sum_{-c_m}\left\{p(c_{m+1}\mid c_m)p(c_m)\prod_k^K p(s_{k,m}\mid c_m)\cdot\beta(c_m)\cdot\prod_k\bar{\nu}_{k,m}^{in}(s_{k,m})\right\}\end{aligned}$$

同样，$\alpha(c_m),\beta(c_m)$的确切值是无法求解的，但是这并不影响输出，我们定义归一化的$\bar{\alpha}(c_m),\bar{\beta}(c_m)$:

$$\begin{aligned}\bar{\alpha}(c_m)&=\frac{\alpha(c_m)}{\sum_{c_m}\alpha(c_m)}\\\bar{\beta}(c_m)&=\frac{\beta(c_m)}{\sum_{c_m}\beta(c_m)}\end{aligned}$$

最终，Module B传递给Module A的消息 $\nu_{h\to s_{k,m}}=\nu_{k,m}^{out}$,

$$\begin{aligned}v_{k,m}^{out}&=\delta_{k,m}\triangleq v_{g_m\to s_{k,m}}\\&\propto\sum_{-s_{k,m}}\left\{g_m\left(c_{m+1},c_m,s_{1,m},\ldots,s_{K,m}\right)\cdot\bar{\alpha}\left(c_{m-1}\right)\bar{\beta}\left(c_m\right)\prod_{-k}\bar{\nu}_{k,m}^{in}(s_{k,m})\right\}\end{aligned}$$

同理，$\bar{\nu}_{h\to s_{k,m}}=\bar{\nu}_{k,m}^{out}$:

$$\bar{\nu}_{k,m}^{out}(s_{k,m})=\frac{\nu_{k,m}^{out}(s_{k,m})}{\sum_{s_{k,m}}\nu_{k,m}^{out}(s_{k,m})}$$

同时，根据Factor Graph 的定义，module B提供给A的$s_{k,m}$的先验概率可以表示为$\pi_{k,m}(s_{k,m})\propto\nu_{k,m}^{out}(s_{k,m})$:

$$\pi_{k,m}(s_{k,m})=\bar{\nu}_{k,m}^{out}(s_{k,m})$$

Update $\boldsymbol{x}_k$

update for $q(\boldsymbol{x}) \triangleq \prod_{k=1}^{K} q_k(\boldsymbol{x}_k)$

$$q_{\boldsymbol{x}_k}(\boldsymbol{x}_k) = \mathcal{CN}(\boldsymbol{x}_k; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

那么本轮计算过后$q^*_{\boldsymbol{x}_k}(\boldsymbol{x}_k)$中的参数更新为：

$$\boldsymbol{\Sigma}_k = \left(\operatorname{diag}\left(\left\langle \frac{\widetilde{a}_{k,1}}{\tilde{b}_{k,1}}, \ldots, \frac{\widetilde{a}_{k,M}}{\tilde{b}_{k,M}} \right\rangle \right) + \mathbf{F}_k^H \operatorname{diag}(\boldsymbol{\kappa}) \mathbf{F}_k \right)^{-1}$$

$$\boldsymbol{\mu}_k = \boldsymbol{\Sigma}_k \mathbf{F}_k^H \operatorname{diag}(\boldsymbol{\kappa}) \boldsymbol{y}_k$$

接下来开始给出出处：

$$\begin{aligned} \ln p(\boldsymbol{x}_k) &\propto \int \ln p(\boldsymbol{v}, \boldsymbol{y}; \boldsymbol{\xi}) d\boldsymbol{\kappa} d\boldsymbol{\gamma} d\boldsymbol{s} d\boldsymbol{c} \prod_{-\boldsymbol{x}_k} d\boldsymbol{x}_{k'} \\ &\propto \int \ln p(\boldsymbol{x} \mid \boldsymbol{\gamma}) \prod_{-\boldsymbol{x}_k} d\boldsymbol{x}_{k'} d\boldsymbol{\gamma} \ + \end{aligned}$$

Update for $\boldsymbol{\gamma}_k$

update for $q(\boldsymbol{\gamma}) \triangleq \prod_{k=1}^{K} q(\boldsymbol{\gamma}_k)$

$$q(\boldsymbol{\gamma}_k) = \prod_{m=1}^{M} \Gamma(\gamma_{k,m}; \widetilde{a}_{k,m}, \tilde{b}_{k,m})$$

Shape parameters:

$$\begin{aligned} \widetilde{a}_{k,m} &= \langle s_{k,m} \rangle a_{k,m} + \langle 1 - s_{k,m} \rangle \bar{a}_{k,m} + 1 \\ &= \widetilde{\pi}_{k,m} a_{k,m} + (1 - \widetilde{\pi}_{k,m}) \bar{a}_{k,m} + 1 \end{aligned}$$

rate parameters：

$$\begin{aligned} \tilde{b}_{k,m} &= \left\langle |x_{k,m}|^2 \right\rangle + \langle s_{k,m} \rangle b_{k,m} + \langle 1 - s_{k,m} \rangle \bar{b}_{k,m} \\ &= |\mu_{k,m}|^2 + \Sigma_{k,m} + \widetilde{\pi}_{k,m} b_{k,m} + (1 - \widetilde{\pi}_{k,m}) \bar{b}_{k,m} \end{aligned}$$

Update $s_{k,m}$

$$q(\boldsymbol{s}_k) = \prod_{m=1}^{M} (\tilde{\pi}_{k,m})^{s_{k,m}} (1 - \tilde{\pi}_{k,m})^{1 - s_{k,m}}$$

Update:

$$\tilde{\pi}_{k,m} = \frac{1}{C} \frac{\pi_{k,m} b_{k,m}^{a_{k,m}}}{\Gamma(a_{k,m})} e^{(a_{k,m}-1)\langle \ln \gamma_{k,m} \rangle - b_{k,m} \langle \gamma_{k,m} \rangle}$$

Where C is the normalization parameter

$$\begin{aligned} C = & \frac{\pi_{k,m} b_{k,m}^{a_{k,m}}}{\Gamma(a_{k,m})} e^{(a_{k,m}-1)\langle \ln \gamma_{k,m} \rangle - b_{k,m} \langle \gamma_{k,m} \rangle} + \\ & \frac{(1 - \pi_{k,m}) \bar{b}_{k,m}^{\bar{a}_{k,m}}}{\Gamma(\bar{a}_{k,m})} e^{(\bar{a}_{k,m}-1)\langle \ln \gamma_{k,m} \rangle - \bar{b}_{k,m} \langle \gamma_{k,m} \rangle} \end{aligned}$$

Where,

$$\langle \ln \gamma_{k,m} \rangle = \psi\left(\tilde{a}_{k,m}\right) - \ln\left(\tilde{b}_{k,m}\right)$$

$\psi(x) \triangleq \frac{d}{dx}\ln(\Gamma(x))$, is the digamma function

```
from scipy.special import gamma
from scipy.special import digamma
```

To reduce the computational complexity, set $a_{k,m} = \bar{a}_{k,m} = 1$

Appendex

$$\boldsymbol{y}_{l,k} = \boldsymbol{\Phi}^H \mathbf{V}\left(\omega_l\right) \mathbf{D}_M\left(\Delta\boldsymbol{\varphi}_k\right) \boldsymbol{x}_k + \mathbf{N}_l, \forall l \in \{1, \ldots, L\}$$

$\boldsymbol{\xi} \triangleq \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \boldsymbol{\xi}_3\}$

$$\begin{aligned}
\boldsymbol{\xi}_1 &= \{\omega_1, \ldots, \omega_L\} \\
\boldsymbol{\xi}_2 &= \{\Delta\varphi_1, \ldots, \Delta\varphi_M\} \\
\boldsymbol{\xi}_3 &= \{\lambda^c, p_{01}^c, p_{10}^c, \mu_1^s, \sigma_1^s, \ldots, \mu_k^s, \sigma_k^s\}
\end{aligned}$$

包含隐变量的联合概率：

$$\begin{aligned}
p(\boldsymbol{v}, \boldsymbol{y}; \boldsymbol{\xi}) &= p(\boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\gamma}, \boldsymbol{s}, \boldsymbol{c}, \boldsymbol{\kappa}) \\
&= p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\kappa}; \boldsymbol{\xi}) p(\boldsymbol{x} \mid \boldsymbol{\gamma}) p(\boldsymbol{\kappa}) p(\boldsymbol{\gamma} \mid \boldsymbol{s}) p(\boldsymbol{c}, \boldsymbol{s}; \boldsymbol{\xi}) \\
&= \underbrace{p(\boldsymbol{x} \mid \boldsymbol{\gamma}) p(\boldsymbol{\kappa}) p(\boldsymbol{\gamma} \mid \boldsymbol{s})}_{\text{known distribution}} \underbrace{p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\kappa}; \boldsymbol{\xi}) p(\boldsymbol{c}, \boldsymbol{s}; \boldsymbol{\xi})}_{\text{with unknown valuables}}
\end{aligned}$$

条件概率：

$$p\left(\boldsymbol{y}_k \mid \boldsymbol{x}_k; \boldsymbol{\xi}\right) = CN\left(\boldsymbol{y}_k; \mathbf{F}_k \boldsymbol{x}_k, \operatorname{Diag}\left(\boldsymbol{\kappa}_k\right)^{-1}\right)$$

$$p(\boldsymbol{y} \mid \boldsymbol{x}; \boldsymbol{\xi}) = \prod_{k=1}^{K} p\left(\boldsymbol{y}_k \mid \boldsymbol{x}_k; \boldsymbol{\xi}\right)$$

$$\begin{aligned}
p(\boldsymbol{c}, \boldsymbol{s}; \boldsymbol{\xi}) &= p(\boldsymbol{c}) \prod_{k=1}^{\mathrm{n}} p\left(\boldsymbol{s}_k \mid \boldsymbol{c}\right) \\
&= p\left(c_1\right) \prod_{k=1}^{K} p\left(s_{k,1} \mid c_1\right) \prod_{m=2}^{M} \left[p\left(c_m \mid c_{m-1}\right) \prod_{k=1}^{K} p\left(s_{k,m} \mid c_m\right) \right]
\end{aligned}$$

Factor Graph

Factor graph - Wikipedia

A factor graph is a bipartite graph representing the factorization of a function. Given a factorization of a function $g(X_1, X_2, \ldots, X_n)$

$$g\left(X_1, X_2, \ldots, X_n\right) = \prod_{j=1}^{m} f_j\left(S_j\right)$$

where $S_j \subseteq \{X_1, X_2, \ldots, X_n\}$, the corresponding factor graph $G = (X, F, E)$ consists of variable vertices $X = \{X_1, X_2, \ldots, X_n\}$, factor vertices $F = \{f_1, f_2, \ldots, f_m\}$, and edges E. The edges depend on the factorization as follows: there is an undirected edge between factor vertex f_j and variable vertex X_k if $X_k \in S_j$. The function is tacitly assumed to be realvalued: $g(X_1, X_2, \ldots, X_n) \in \mathbb{R}$

Factor graphs can be combined with message passing algorithms to efficiently compute certain characteristics of the function $g(X_1, X_2, \ldots, X_n)$, such as the marginal distributions.

message passing algorithms are usually exact for trees, but only approximate for graphs with cycles.

Message passing

Variational message passing - Wikipedia

> **Variational message passing** (**VMP**) is an approximate inference technique for continuous- or discrete-valued Bayesian networks, with conjugate-exponential parents, developed by John Winn. VMP was developed as a means of generalizing the approximate variational methods used by such techniques as Latent Dirichlet allocation and works by updating an approximate distribution at each node through messages in the node's Markov blanket.

定义观测变量$\boldsymbol{y}$,隐变量$\boldsymbol{v}$,

首先有观测变量本身的似然函数:

$$\begin{aligned}\ln p(\boldsymbol{y}) &= \int_v q(\boldsymbol{v}) \ln\left(\frac{p(\boldsymbol{v}, \boldsymbol{y})}{p(\boldsymbol{v} \mid \boldsymbol{y})}\right) \\ &= \int_v q(\boldsymbol{v}) \left[\ln \frac{p(\boldsymbol{v}, \boldsymbol{y})}{q(\boldsymbol{v})} - \ln \frac{p(\boldsymbol{v} \mid \boldsymbol{y})}{q(\boldsymbol{v})}\right] \\ &= \int_v q(\boldsymbol{v}) \ln \frac{p(\boldsymbol{v}, \boldsymbol{y})}{q(\boldsymbol{v})} + \underbrace{(-1) \int_v q(\boldsymbol{v}) \ln q(\boldsymbol{v})}_{\text{relative entropy (non-negative)}}\end{aligned}$$

所以有$\ln p(\boldsymbol{y})$ 的下界(ELBO):

$$\int_v q(\boldsymbol{v}) \ln \frac{p(\boldsymbol{v}, \boldsymbol{y})}{q(\boldsymbol{v})}$$

当$q(\boldsymbol{v})$ 可以被分解为:

$$q(\boldsymbol{v}) = \prod_i q_i(\boldsymbol{v}_i)$$

where $\boldsymbol{v}_i$ is a disjoint part of the graphical model

This is the key: We can maximize ELOB, or $\mathcal{L}(q)$, by minimizing this special KL divergence, where we can find approximate and optimal $q_i^*(\boldsymbol{v}_i)$, such that:

$$\ln q_i^*(\boldsymbol{v}_i) = E_{-\boldsymbol{v}_i}[\ln p(\boldsymbol{v}, \boldsymbol{y})]$$

Sum-product rule

[Belief propagation - Wikipedia]

It calculates the [marginal distribution] for each unobserved node (or variable), conditional on any observed nodes (or variables).

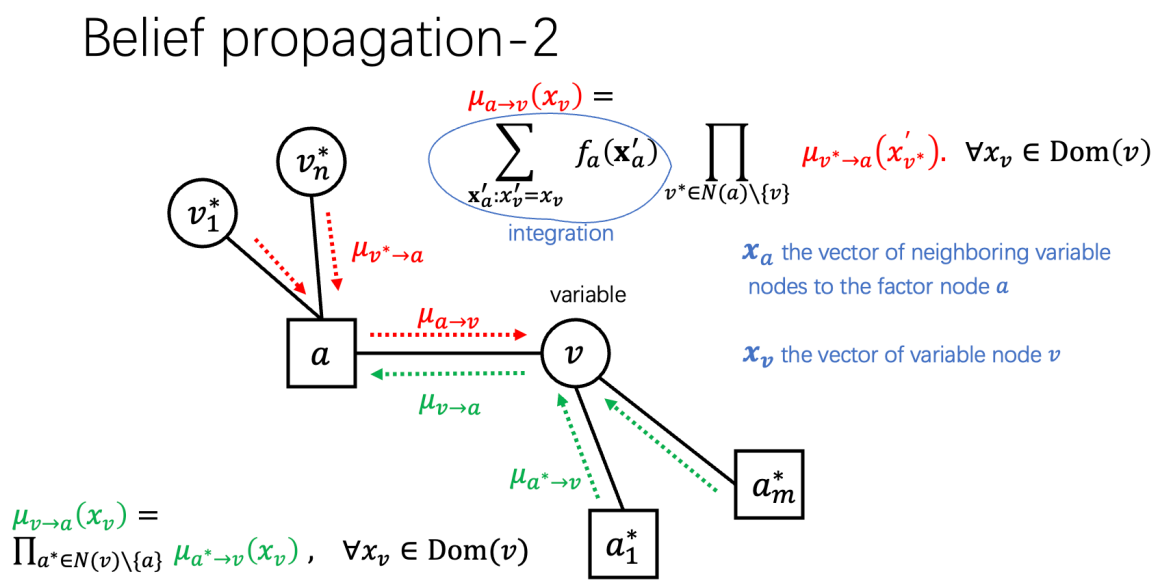
$$p(\mathbf{x}) = \prod_{a \in F} f_a(\mathbf{x}_a)$$

The algorithm works by passing real valued functions called ***messages*** along with the edges between the hidden nodes.

More precisely, if *v* is a variable node and *a* is a factor node connected to *v* in the factor graph, the messages from *v* to *a*, (denoted by $\mu_{v\to a}$and from *a* to *v* ($\mu_{a\to v}$), are real-valued functions whose domain is Dom(*v*), the set of values that can be taken by the random variable associated with *v*. These messages contain the "influence" that one variable exerts on another. The messages are computed differently depending on whether the node receiving the message is a variable node or a factor node. Keeping the same notation:

$$\forall x_v \in \mathrm{Dom}(v), \mu_{v\to a}(x_v) = \prod_{a^* \in N(v)\setminus\{a\}} \mu_{a^*\to v}(x_v)$$

$$\forall x_v \in \mathrm{Dom}(v), \mu_{a\to v}(x_v) = \sum_{\mathbf{x}'_a : x'_v = x_v} f_a(\mathbf{x}'_a) \prod_{v^* \in N(a)\setminus\{v\}} \mu_{v^*\to a}(x'_{v^*})$$



upon convergence

1. 每个v节点的发生概率有：

$$p_{X_v}(x_v) \propto \prod_{a \in N(v)} \mu_{a\to v}(x_v)$$

2. 每个函数节点的输出概率有：

$$p_{X_a}(\mathbf{x}_a) \propto f_a(\mathbf{x}_a) \prod_{v \in N(a)} \mu_{v\to a}(x_v)$$

当概率图为tree的时候，可以分部更新，但是当概率图含有loop时，就不一定收敛

varitional bayes

$$\ln p(\boldsymbol{y}) = \underbrace{D_{KL}(q\|p)}_{\text{target: minimization}} + \underbrace{\mathcal{L}(q)}_{\text{equivalence: maximization}}$$

ELBO：

$$\mathcal{L}(q) = \int_{\boldsymbol{v}} q(\boldsymbol{v}) \ln \frac{p(\boldsymbol{v}, \boldsymbol{y})}{q(\boldsymbol{v})}$$

Mean field approximation

Factorize $q(\boldsymbol{v})$

$$q(\boldsymbol{v}) = \prod_{i=1}^{\mathcal{Q}} q_i(\boldsymbol{v}_i \mid \boldsymbol{y})$$

$$\ln q_j^*(\boldsymbol{v}_j) = \mathrm{E}_{-\boldsymbol{v}_j}\left[\ln p(\boldsymbol{v}, \boldsymbol{y})\right] + \text{constant}$$

the expectation $\mathrm{E}_{-\boldsymbol{v}_j}\left[\ln p(\boldsymbol{v}, \boldsymbol{y})\right]$ can usually be simplified into a function of the fixed hyperparameters of the prior distributions over the latent variables and of expectations (and sometimes higher moments such as the variance) of latent variables not in the current partition (i.e. latent variables not included in

New basic

The variational approximation for Bayesian inference | IEEE Journals & Magazine | IEEE Xplore

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} p(\mathbf{x}; \boldsymbol{\theta}) \tag{1}$$

$p(\mathbf{x}; \boldsymbol{\theta})$ is usually impossible to compute directly

此时引入隐变量$\mathbf{z}$

they brought enough information for oobservations so that $p(\mathbf{x} \mid \mathbf{z})$ is easy to solve

那么，就可以通过边缘概率求解原来难以解决的$p(\mathbf{x}; \theta)$

$$p(\mathbf{x}; \boldsymbol{\theta}) = \int p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) d\mathbf{z} = \int p(\mathbf{x} \mid \mathbf{z}; \boldsymbol{\theta}) p(\mathbf{z}; \boldsymbol{\theta}) d\mathbf{z} \tag{2}$$

如果上式可以解决，那么隐变量的后验概率也可以得到：

$$p(\mathbf{z} \mid \mathbf{x}; \boldsymbol{\theta}) = \frac{p(\mathbf{x} \mid \mathbf{z}; \boldsymbol{\theta}) p(\mathbf{z}; \boldsymbol{\theta})}{p(\mathbf{x}; \boldsymbol{\theta})} \tag{3}$$

尽管(3)中的形式看起来很简单，但是在一般情况下(2)中的积分都是不可解的。因此，接下来的目的在于绕过(2)中的积分

绕过的方式有两种主要的大类：

1. Monte Carlo 方法
2. deterministic approximations

3. maximum posteriori （MAP）

   is an extention of ML

VBI -> approximate posterior

Illustration of EM algthrithm

$$\ln p(\mathrm{x};\boldsymbol{\theta}) = F(q,\boldsymbol{\theta}) + KL(q\|p)$$

ELBO:

$$F(q,\boldsymbol{\theta}) = \int q(\mathbf{z}) \ln\left(\frac{p(\mathbf{x},\mathbf{z};\boldsymbol{\theta})}{q(\mathbf{z})}\right) d\mathbf{z}$$

KLD:

$$\mathrm{KL}(q\|p) = -\int q(\mathbf{z}) \ln\left(\frac{p(\mathbf{z}\mid\mathbf{x};\boldsymbol{\theta})}{q(\mathbf{z})}\right) d\mathbf{z}$$

Problem formulation:

$$\max_{q,\theta} F(q,\boldsymbol{\theta})$$

EM framework:

$$\begin{aligned} &\text{E-step}: \quad \text{Compute} \quad p\left(\mathbf{z}\mid\mathbf{x};\boldsymbol{\theta}^{\mathrm{OLD}}\right) \\ &\text{M-step}: \quad \text{Evaluate} \quad \boldsymbol{\theta}^{\mathrm{NEW}} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta},\boldsymbol{\theta}^{\mathrm{OLD}}\right) \end{aligned}$$

Obviercely, EM framework requires that the posterior $p(\mathbf{z}\mid\mathbf{x};\boldsymbol{\theta})$ is explicitly know or at least able to compute the integration $\langle \ln p(\mathbf{z}\mid\mathbf{x};\boldsymbol{\theta})\rangle_{p(\mathbf{z}|\mathbf{x};\boldsymbol{\theta}^{\mathrm{OLD}})}$

Variational EM

一般来讲，估计$q(\mathbf{z})$，需要先假设其公式已知，然后推导其参数$\boldsymbol{\omega}$，即，先写为$q(\mathbf{z};\boldsymbol{\omega})$

那么问题就化简为：

$$\max_{\boldsymbol{\omega},\theta} F(\boldsymbol{\omega},\boldsymbol{\theta})$$

这样的变分估计有一个很成功的变种：$\mathrm{factorized\ approximation}$

$$q(\mathbf{z}) = \prod_{i=1}^{M} q_i(z_i)$$

那么将ELBO写为：

$$
\begin{aligned}
F(q, \boldsymbol{\theta}) &= \int \prod_i q_i \Big[\ln p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) - \sum_i \ln q_i \Big] d\mathbf{z} \\
&= \int \prod_i q_i \ln p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) \prod_i dz_i - \sum_i \int \prod_j q_j \ln q_i dz_i \\
&= \int q_j \left[\ln p\left(\mathrm{x}, \mathrm{z}; \theta\right) \prod_{i \neq j} \left(q_i dz_i\right) \right] dz_j - \int q_j \ln q_j dz_j - \sum_{i \neq j} \int q_i \ln q_i dz_i \\
&= \int q_j \ln q_j dz_j - \sum_{i \neq j} \int q_i \ln q_i dz_i \\
&\quad - \sum_{i \neq j} \int q_i \ln q_i dz_i \\
&= -\mathrm{KL}\left(q_j \| \tilde{p}\right) - \sum_{i \neq j} \int q_i \ln q_i dz
\end{aligned}
$$

Where:

$$
\ln \tilde{p}\left(\mathbf{x}, z_j; \boldsymbol{\theta}\right) = \left\langle \ln p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) \right\rangle_{i \neq j} = \int \ln p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) \prod_{i \neq j} \left(q_i dz_i\right)
$$

最优情况是KLD为0:

$$
\ln q_j^*\left(z_j\right) = \left\langle \ln p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) \right\rangle_{i \neq j} + \text{ const.}
$$

$$
q_j^*\left(z_j\right) = \frac{\exp\left(\left\langle \ln p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) \right\rangle_{i \neq j}\right)}{\int \exp\left(\left\langle \ln p(\mathbf{x}, \mathbf{z}; \boldsymbol{\theta}) \right\rangle_{i \neq j}\right) dz_j}
$$

New basic - Factor graph

[Factor graphs and the sum-product algorithm | IEEE Journals & Magazine | IEEE Xplore]

Variables: $x_1, x_2, \ldots, x_n$

domain (or alphabet) $A_1, A_2, \ldots, A_n$

R-valued function $g(x_1, x_2, \ldots, x_n)$

domain of function $g(x_1, x_2, \ldots, x_n)$, $S\ (configuration\ space)$

$$
S = A_1 \times A_2 \times \cdots \times A_n
$$

$codomain\ R$

"not sum" concept

$g(x_1, x_2, \ldots, x_n)$ Factors into a product of several $local\ functions$

$$
g(x_1, x_2, \ldots, x_n) = \prod_{j \in \mathcal{J}} f_j(X_j)
$$

Then a $factor\ graph$ can be given to illustrate the construction of factorlization of g

in many cases, we are interested in computing the marginal probability $g_i(x_i)$

The key : when a factor graph is cycle-free, the factor graph not only encodes in its structure the factorization of the global function, but also encodes arithmetic expressions by which the marginal functions associated with the global func□tion may be computed.

\ding1

\circled1

---

1. Factor_graphs_and_the_sum-product_algorithm