【2022-09-04】-【20230317】

Local Variational Methods
局部变分下界的构造
应用：Variational Logistic Regression
Expectation Propagation
EP 算法的思路与推导
EP 算法在 graphs 上的应用：message passing
EP 算法在概率图模型上的推导
从 EP 到 BP：message passing

Variational Inference 两种角度的扩展，一种是从全局近似到局部近似，一种是不同形式的 KL 散度。

Local Variational Methods

局部变分下界的构造

对应 10.5 Local Variational Methods 小节，讨论了如何构造凸函数的下界，或者说 concave (凹的) 函数的上界，本质上就是函数的切线。

> a (strictly) convex (凸的) function: every chord lies above the function.

convex function $f(x)$，任意一点 ξ 的切线为 $f'(\xi)x + f(\xi)$，则

$$f(x) \geqslant f'(\xi)(x-\xi) + f(\xi)$$

记 $\eta = f'(\xi)$，则上述不等式可改写为

$$f(x) \geqslant \eta x - g(\eta)$$

其中 $g(\eta)$ 的函数表达式可以从 $\eta = f'(\xi), g(\eta) = f'(\xi)\xi - f(\xi)$ 的关系中计算得到。此外，基于上述不等关系，$g(\eta)$ 还可以通过求解如下的优化问题得到：

$$g(\eta) = \max_x \{\eta x - f(x)\}$$

事实上，为求解上述优化问题，我们对 x 求导，令导数等于 0，并记 x 的最优解为 ξ，即得 $\eta = f'(\xi)$，回代可得 $g(\eta) = f'(\xi)\xi - f(\xi)$，可以看到与前文等价。

We have succeeded in approximating the convex function $f(x)$ by a simpler, linear function $y(x,\eta) \triangleq \eta x - g(\eta)$. The price we have paid is that we have introduced a variational parameter η, and to obtain the tightest bound we must optimize with respect to η，即 $g(\eta) = \max_x \{\eta x - f(x)\}$.

前面是已知凸函数 $f(x)$，求 $g(\eta)$。事实上，我们也可以已知 $g(\eta)$ 求 $f(x)$，即已知凸函数任意一点的切线方程，求原凸函数，即对任意给定的点x，其在切线上的值为 $\eta x - g(\eta)$，当滑动切线恰好为 x 处时，$\eta x - g(\eta)$ 最大，且与 $f(x)$ 相等，则：

$$f(x) = \max_\eta \{\eta x - g(\eta)\}$$

综上，已知 $f(x)$ 求 $g(\eta)$，已知 $g(\eta)$ 求 $f(x)$ 分别为：

$$g(\eta) = \max_x \{\eta x - f(x)\}$$
$$f(x) = \max_\eta \{\eta x - g(\eta)\}$$

也称其为 **convex duality**. 类似地，**concave duality**：

$$g(\eta) = \min_x \{\eta x - f(x)\}$$
$$f(x) = \min_\eta \{\eta x - g(\eta)\}$$

若函数不具备凹凸性，则可通过变量（包括自变量与因变量）代换使其具备凹凸性，得到变分下界后再变换回来即可：If the function of interest is not convex (or concave), then we cannot directly apply the method above to obtain a bound. However, we can first seek invertible transformations either of the function or of its argument which change it into a convex form. We then calculate the conjugate function and then transform back to the original variables.

文中以 Sigmoid 函数为例分别构造了其 local 变分上界和变分下界，但相关结果无法直接推广到 Softmax 函数中。此外，构造的 Sigmoid 的 local 下界 has the form of the exponential of a quadratic function of x，因此可用于 seek Gaussian representations of posterior distributions defined through logistic sigmoid functions.

与 global variational approach 的对比

1. 变分下界的构造都基于函数的凹凸性，global 利用的是对数函数的凹凸性，这使得 KL 散度项非负，从而得到变分下界：

$$\ln p(\mathrm{X}) \geqslant \ln p(\mathrm{X}) - \mathrm{KL}(q \| p(\mathrm{Z}|\mathrm{X})) = \sum_{\mathrm{Z}} q(\mathrm{Z}) \ln \frac{p(\mathrm{X}, \mathrm{Z})}{q(\mathrm{Z})} \triangleq \mathcal{L}(p, q)$$

2. global vs local：
    1. 首先 global 下界 $\mathcal{L}(p, q)$ 通过选择合适的变分函数 $q(\mathrm{Z}) = p(\mathrm{Z}|\mathrm{X})$ 可以实现对原函数 $\ln p(X)$ 的 global 相等，而 local 下界 $y(x, \eta) = \eta x - g(\eta)$ 无论如何选择变分参数 η，都只是与 $f(x)$ 局部相等（切点处），但做不到处处相等。
    2. 可以看到，global 下界与 local 下界处理的也是不同的问题，而不是一个问题的两种处理方法，前者构造的是 $\ln p(\mathrm{X})$ 的变分下界，而后者构造的是凸函数 $f(x)$ 的变分下界。
    3. global 变分下界引入的是变分函数 $q(\mathrm{Z})$，local 变分下界引入的是变分参数 η。注意，$g(\eta)$ 是关于变分参数 η 的函数，而不是变分函数，在给定凸函数 $f(x)$ 后 $g(\eta)$ 就固定下来了，是已知的。

应用：Variational Logistic Regression

对应 10.6 Variational Logistic Regression 小节。

The variational framework discussed in Sections 10.1 and 10.2 can be considered a ‘global’ method in the sense that it directly seeks an approximation to the full posterior distribution over all random variables.

An alternative ‘local’ approach involves finding bounds on functions over individual variables or groups of variables within a model. For instance, we might seek a bound on a conditional distribution $p(y|x)$, which is itself just one factor in a much larger probabilistic model specified by adirected graph. The purpose of introducing the bound of course is to simplify the resulting distribution. 注意，变分推断 (全局变分) 与局部变分并不是替代关系，它们可以一起使用，比如 10.6.3 Inference of hyperparameters 小节，因此这里的 'alternative' 应理解为'可供选择的'。此外，'over individual variables or groups of variables within a model' 可能会误导我们对 local 的理解，以为局部变分的”局部“是指只能针对几个变量而不是针对所有变量。显然，我们可以直接构造整个函数 $f(\mathrm{x})$ 的局部变分下界，但若 $f(\mathrm{x}) = \prod f_i(\mathrm{x}_i)$，我们也可以针对 factor $f_i(\mathrm{x}_i)$ 构造变分下界，10.6.1

Variational posterior distribution 和 10.6.2 Optimizing the variational parameters 小节就才用了这样的方案。不论怎样，引入下界的目的都是为了简化目标分布，使之便于计算。

注意，不要被书中的标题所迷惑了，10.6.1、10.6.2 两小节与 10.6.3 小节不是承接关系，是两个独立的体系，它们推导时基于的假设都不一样。此外，10.6.3 小节为了完成对超参的优化，实际上也讨论了前面两小节的主题，可标题只写了超参的优化，这就让人误以为是承接 10.6.1 与 10.6.2 小节，但事实上 10.6.3 小节自成一个完备的体系。

只用了局部变分进行近似：

10.6.1 Variational posterior distribution 基于 10.5 小节得到的 sigmoid 函数的局部变分下界即式 (10.144)，引入变分参数 ξ_n，构造 $p(t_n|\mathrm{w})$ 的下界，进而对整个数据集 $\mathbf{t}$，可得 $p(\mathbf{t},\mathrm{w})=\prod_{n=1}^{N}p(t_n)p(\mathrm{w})$ 的变分下界。接下来，用变分下界作为原分布的近似，计算参数后验 $p(\mathrm{w}|\mathbf{t})$ 的近似分布 $q(\mathrm{w})$.

10.6.2 Optimizing the variational parameters 小节则在 10.6.1 小节的基础上优化变分参数 ξ_n ($n=1,\cdots,N$)，实现最佳的近似。

全局变分 (变分推断) + 局部变分：

10.6.3 Inference of hyperparameters 小节进行了2次近似，在变分推断的大框架下，又嵌入了局部变分进行近似。

Expectation Propagation

对应 10.7 Expectation Propagation 小节。

EP 算法的思路与推导

we call that, 变分推断(Variational Inference, VI)/变分贝叶斯(Variational Bayes, VB)实际上就是

$$\min_q \mathrm{KL}_q(q\|p)$$

- 虽然目标分布 p 的函数形式已知，但其积分 intractable，因此归一化常数未知；
- 我们用一个更简单的函数 q 去近似 p，且通常对近似分布 q 只作平均场假设；
- p 可以是后验分布，也可以是其他任意待近似的分布，只不过当 p 是后验分布时，就存在 ELBO 那一套话术了。

> 注意，EM 算法与 VI 并不相同，它不是简单的 $\min_q \mathrm{KL}(q\|p)$，EM 将隐变量 Z 和参数 θ 分开处理，因此还涉及到 E 步 θ 的优化，而 VB/VI 则可视为全部在 M 步中搞定。更多细节参见上一篇笔记 6.1 。

As with the variational Bayes methods discussed so far, *expectation propagation* (EP) too is based on the minimization of a Kullback-Leibler divergence but now of the **reverse** form, which gives the approximation rather different properties. 即 EP 也是基于 KL 散度用 q 去近似目标分布 p，但和 VI 不同的是，KL 散度中 p,q 的位置互换了：

$$\min_q \mathrm{KL}(p\|q)$$

同样地，p 可以是后验分布，也可以是任意待近似的分布。

下面我们开始介绍 EP 算法，我们只点出重点和思路，推导细节见书本。

已知分布

$$p(\mathcal{D}, \boldsymbol{\theta}) = \prod_i f_i(\boldsymbol{\theta}) \tag{10.188}$$

其中，$\boldsymbol{\theta}$ 表示隐变量 + 参数 (与介绍 VI 时的 $\mathbb{Z}$ 含义相同)；$\mathcal{D}$ 表示可观测数据，是已知的，因此式 (10.188) 右边并未显示地标出。注意，函数 f_i 均已知，但是，若我们想知道 $p(\boldsymbol{\theta}|\mathcal{D}) = p(\mathcal{D}, \boldsymbol{\theta})/p(\mathcal{D})$ or model evidence $p(\mathcal{D})$ 我们需要计算

$$p(\mathcal{D}) = \int \prod_i f_i(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}$$

而这一积分我们假设是 intractable 的。为此，我们尝试构造目标分布 $p(\boldsymbol{\theta}|\mathcal{D})$ 的近似分布 $q(\boldsymbol{\theta})$，并基于更简单的 $q(\boldsymbol{\theta})$ 进行一系列的推断。注意，目标分布不一定要是后验分布 (形式表现为条件分布，即存在观测数据 $\mathcal{D}$)，正如前面所说，它可以为任意的概率分布：

$$p(\boldsymbol{\theta}) = \frac{1}{C} \prod_i f_i(\boldsymbol{\theta})$$

$$C = \int \prod_i f_i(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}$$

函数 f_i 均为已知，但计算 C 的积分 intractable，前面的 $p(\mathcal{D})$ 就对应这里的归一化常数 C。下面，我们还是和原文一样使用 $p(\boldsymbol{\theta}|\mathcal{D})$ 的形式介绍 EP 算法。

首先取近似分布 $q(\boldsymbol{\theta})$ 为 $p(\boldsymbol{\theta}|\mathcal{D})$ 相似的形式：

$$q(\boldsymbol{\theta}) = \frac{1}{Z} \prod_i \widetilde{f}_i(\boldsymbol{\theta})$$

并一一对应，用 $\widetilde{f}_i$ 去近似 f_i 。注意，这种近似相差一个常数因子，即 $\widetilde{f}_i \approx c f_i$，因为反正最后都会通过 Z 归一化，所以并不要求在绝对大小上相近，可以理解为形状相似，但对大小不作要求。EP 理论上尝试最小化：

$$\mathrm{KL}(p\|q) = \mathrm{KL}\left(\frac{1}{p(\mathcal{D})} \prod_i f_i(\boldsymbol{\theta}) \middle\| \frac{1}{Z} \prod_i \widetilde{f}_i(\boldsymbol{\theta}) \right)$$

但 In general, this minimization will be intractable because the KL divergence involves averaging with respect to the true distribution.

一种方案是分别最小化每个 $f_i(\boldsymbol{\theta})$ 和 $\widetilde{f}_i(\boldsymbol{\theta})$ 对的 KL 散度，这样求解很简单，也无需迭代计算。但由于所有因子都是单独近似的，这并不能保证累乘后整体的近似效果。As a rough approximation, we could instead minimize the KL divergences between the corresponding pairs $f_i(\boldsymbol{\theta})$ and $\widetilde{f}_i(\boldsymbol{\theta})$ of factors. This represents a much simpler problem to solve, and has the advantage that the algorithm is noniterative. However, because each factor is individually approximated, the product of the factors could well give a poor approximation.

为此，EP采用如下方案：Expectation propagation makes a much better approximation by optimizing each factor in turn in the context of all of the remaining factors，即不断轮流迭代更新每个 factor，以更新 factor $\widetilde{f}_j(\boldsymbol{\theta})$ 为例 (为了以示区分，我们记更新的因子为 $\widetilde{f}_j'(\boldsymbol{\theta})$)，我们最小化：

$$\min_{\widetilde{f}_j} \mathrm{KL}\left(\frac{f_j(\boldsymbol{\theta})q^{\setminus j}(\boldsymbol{\theta})}{Z_j}\middle\|\frac{\widetilde{f}_j'(\boldsymbol{\theta})q^{\setminus j}(\boldsymbol{\theta})}{K_j}\right)$$

和书中不同，我们这里只要求取 $q^{\setminus j}(\boldsymbol{\theta})$ 正比于 $\prod_{i\neq j}\widetilde{f}_i(\boldsymbol{\theta})$ 即可，反正最后都会通过 $Z_j=\int f_j(\boldsymbol{\theta})q^{\setminus j}(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$ 归一化，书中基于 $q^{\setminus j}(\boldsymbol{\theta})=q(\boldsymbol{\theta})/\widetilde{f}_j(\boldsymbol{\theta})$ 进行介绍反而还搞复杂了，没有这里简洁和直接。可以看到，更新 $\widetilde{f}_j(\boldsymbol{\theta})$ 时，目标分布 $p(\boldsymbol{\theta}|\mathcal{D})$ 只贡献了一个因子 $f_j(\boldsymbol{\theta})$：

- 这使得各种计算都 intractable，比如 Z_j 的计算；
- 同时为了达到 $\min_q \mathrm{KL}(p\|q)$ 的效果：剩余的 $\prod_{i\neq j}f_i(\boldsymbol{\theta})$ 用 $\prod_{i\neq j}\widetilde{f}_i(\boldsymbol{\theta})$ 替代，并轮流迭代更新，当计算收敛时，即完成了 $\min_q \mathrm{KL}(p\|q)$，这也就是前文所说的 optimizing each factor in turn in the context of all of the remaining factors。

此外，

- $f_j(\boldsymbol{\theta}), q^{\setminus j}(\boldsymbol{\theta}), Z_j$ 已知，未知的是 $\widetilde{f}_j'(\boldsymbol{\theta})$ 与 $K_j=\int \widetilde{f}_j'(\boldsymbol{\theta})q^{\setminus j}(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$。通过最小化上述 KL 散度，我们实际上能确定的量是 $\frac{\widetilde{f}_j'(\boldsymbol{\theta})}{K_j}$，而函数 $\widetilde{f}_j'(\boldsymbol{\theta})$ 与常数 K_j 之间的分配是任意的，即 $\widetilde{f}_j'(\boldsymbol{\theta})$ 是对 $f_j(\boldsymbol{\theta})$ 相差常数因子的近似。但通过令 $K_j=Z_j$，in the context of $\frac{q^{\setminus j}(\boldsymbol{\theta})}{Z_j}$，近似在同一量级，即 $\widetilde{f}_j'(\boldsymbol{\theta})\approx f_j(\boldsymbol{\theta})$。注意，因为是近似，而不是相等，同一量级的近似本身就没有固定标准，需要基于特定语境，比如，我们也可以令 $\int \widetilde{f}_j'(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}=\int f_j(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$ 来实现 $\widetilde{f}_j'(\boldsymbol{\theta})\approx f_j(\boldsymbol{\theta})$。显然，基于不同语境得到的 $\widetilde{f}_j'(\boldsymbol{\theta})$ 并不相同。
- 在得到 $\widetilde{f}_j'(\boldsymbol{\theta})$ 后，我们令 $\widetilde{f}_j(\boldsymbol{\theta})\leftarrow\widetilde{f}_j'(\boldsymbol{\theta})$，并继续优化下一个 factor。与 VI 不同，EP 算法并不保证收敛，但若收敛，那么在完成最后一次更新后，基于最新的 factor，我们有 $p(\mathcal{D})=\int\prod_i f_i(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}\approx\int\prod_i\widetilde{f}_i(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$，这是因为每次更新我们都令 $K_j=Z_j$，这使得近似在同一量级，即 $\widetilde{f}_i(\boldsymbol{\theta})\approx f_i(\boldsymbol{\theta}),\ \text{for all } i$。
- 我们假设上述 KL 散度的优化能很方便地进行，比如，书中假设组成 q 的 factor 来源于指数族分布 (这意味着 q 也为指数族分布)，因此可以直接基于 moment matching 得到 $\widetilde{f}_j'(\boldsymbol{\theta})$ 的解析解。注意，不要误以为 q 为指数族分布是 EP 算法的必要条件，下一小节在介绍 EP 算法在 graphs 上的应用时就并没有要求 q 为指数族分布，此时 KL 散度的优化使用基于平均场假设的解析解，即书中式 (10.17)。

10.7.1 Example: The clutter problem 小节则给出了一个使用 EP 算法求解的例子，其中一些量在更新前后并未作区分，显得很混乱，我们在书中将更新后的量加上上标 $'$ 以示区别。

EP 算法在 graphs 上的应用：message passing

> 对应 10.7.2 Expectation propagation on graphs 小节

EP 算法在概率图模型上的推导

注意，本小节的推导无需假设概率图 tree-structure。

首先介绍前提条件。

We now consider situations in which the factors depend only on subsets of the variables. Such restrictions can be conveniently expressed using the framework of probabilistic graphical models, as discussed in Chapter 8.

$$p(\boldsymbol{\theta})=\prod_i f_i(\boldsymbol{\theta}_i)$$

where $\boldsymbol{\theta}_i$ represents the subset of variables associated with factor f_i. 也就是假设分布 $p(\boldsymbol{\theta})$ 中的 factor 只依赖于部分变量而不是所有变量，即从 $f_i(\boldsymbol{\theta})$ 变为 $f_i(\boldsymbol{\theta}_i)$，由此概率分布可用概率图很方便地描述。

We approximate this (指上面的 $p(\boldsymbol{\theta})$) using a fully factorized distribution of the form

$$q(\boldsymbol{\theta}) \propto \prod_i \widetilde{f}_i(\boldsymbol{\theta}_i) = \prod_i \prod_{\theta_k \in \boldsymbol{\theta}_i} \widetilde{f}_{ik}(\theta_k)$$

where θ_k corresponds to an individual variable node. 对近似分布 q，我们假设其 factor fully factorized，即 $\widetilde{f}_i(\boldsymbol{\theta}_i) = \prod_{\theta_k \in \boldsymbol{\theta}_i} \widetilde{f}_{ik}(\theta_k)$。在概率图中，$i$ 对应因子结点 $\widetilde{f}_i$，而 k 对应与 i 邻接的变量结点 θ_k。

接下来，我们完整走一遍 EP 算法的流程。

首先 remove the term $\widetilde{f}_j(\boldsymbol{\theta}_j)$ from $q(\boldsymbol{\theta})$ to give

$$q^{\backslash j}(\boldsymbol{\theta}) \propto \prod_{i \neq j} \prod_{\theta_k \in \boldsymbol{\theta}_i} \widetilde{f}_{ik}(\theta_k)$$

此时，我们要最小化 KL 散度

$$\min_{\widetilde{f}_j'} \ \mathrm{KL}\left(\frac{f_j(\boldsymbol{\theta}_j) q^{\backslash j}(\boldsymbol{\theta})}{Z_j} \middle\| \frac{\widetilde{f}_j'(\boldsymbol{\theta}_j) q^{\backslash j}(\boldsymbol{\theta})}{K_j} \right)$$

注意，这里并没有假设 q 为指数族分布，因此不能用 moment matching 求解上式。但因为 q 的 fully factorized 假设，我们可以使用基于平均场假设给出的解析解，即式 (10.17) 求解上式。

> 式 (10.17) 给出了在 q 满足平均场假设，即 $q = \prod_i^M q_i(\mathrm{Z}_i)$，其中 $\prod_i^M \mathrm{Z}_i = \mathrm{Z}$ 且 $\mathrm{Z}_i \cap \mathrm{Z}_j = \emptyset, \ \text{for } i \neq j$ 时最小化 KL 散度 $\min_q \mathrm{KL}(p \| q)$ 的解：
>
> $$q_j^\star(\mathrm{Z}_j) = \int p(\mathrm{Z}) \prod_{i \neq j} \mathrm{d}\mathrm{Z}_i \tag{10.17}$$
>
> 即 the optimal solution for $q_j(\mathrm{Z}_j)$ is just given by the corresponding marginal distribution of $p(\mathrm{Z})$. Note that this is a closed-form solution and so does not require iteration.

过程如下，首先记目标分布为 $\widehat{p}(\boldsymbol{\theta}) = \frac{f_j(\boldsymbol{\theta}_j) q^{\backslash j}(\boldsymbol{\theta})}{Z_j}$，由式 (10.17) 可知，$q$ 的解析解：

$$q^\star(\boldsymbol{\theta}) = \prod_m \widehat{p}(\theta_m)$$

$$\text{where } \widehat{p}(\theta_m) = \int p(\boldsymbol{\theta}) \prod_{k \neq m} \mathrm{d}\theta_k = \int p(\boldsymbol{\theta}) \mathrm{d}\{\boldsymbol{\theta} \backslash \theta_m\}$$

则

$$\widetilde{f}_j'(\boldsymbol{\theta}_j) \propto \frac{q^*(\boldsymbol{\theta})}{q^{\backslash j}(\boldsymbol{\theta})}$$

我们强调，各 factor $\widetilde{f}_i$ 的自变量之间并不要求 disjointed，可以包含相同的元素。但由于引入了 fully factorized 假设，factor $\widetilde{f}_i(\boldsymbol{\theta}_i)$ 可进一步分解为只包含一个变量的子 factor，因此 KL 散度右边在单个变量的维度上是满足平均场假设，可以基于式 (10.17) 求解。

下面对上述结果进行化简，首先 $q^{\backslash j}(\boldsymbol{\theta})$:

$$
\begin{aligned}
q^{\backslash j}(\boldsymbol{\theta}) &= \prod_{i\neq j} \widetilde{f}_i(\boldsymbol{\theta}_i) = \prod_{i\neq j}\prod_k \widetilde{f}_{ik}(\theta_k) \\
&= \prod_{i\neq j,k} f_{ik}(\theta_k)
\end{aligned}
$$

$\widehat{p}(\theta_m)$ 则需分情况讨论：

- 当 $\theta_m \in \boldsymbol{\theta}_j$ 时：

  $$
  \widehat{p}(\theta_m) \propto \left(\int f_j(\boldsymbol{\theta}_j) \prod_{n\neq j, \theta_k \in \boldsymbol{\theta}_j \backslash \theta_m} \widetilde{f}_{nk}(\theta_k) \mathrm{d}\{\boldsymbol{\theta}_j \backslash \theta_m\} \right) \prod_{i\neq j} \widetilde{f}_{im}(\theta_m)
  $$

  其中，$\prod_{n\neq j, \theta_k \in \boldsymbol{\theta}_j \backslash \theta_m} \widetilde{f}_{nk}(\theta_k)$ 表示在概率图上那些与因子 j 通过一个非 θ_m 的变量结点 θ_k 相连的子 factor 的累乘，大括号内的积分会积掉所有非 θ_m 的变量，使得自变量只剩 θ_m；
  $\prod_{i\neq j} \widetilde{f}_{im}(\theta_m)$ 表示那些与因子 j 通过变量 θ_m 相连的子 factor 的累乘；

- 当 $\theta_m \notin \boldsymbol{\theta}_j$ 时：

  $$
  \widehat{p}(\theta_m) \propto \prod_i \widetilde{f}_{im}(\theta_m)
  $$

  $\prod_i \widetilde{f}_{im}(\theta_m)$ 表示所有含有 θ_m 的子 factor 的累乘，因为 factor j 不含 θ_m 所以 $\prod_i$ 下标不用 $i \neq j$.

可以看到，将所有 $\widehat{p}(\theta_m)$ 组合到一起，非大括号积分的部分组成了 $q^{\backslash j}(\boldsymbol{\theta})$，因此：

$$
\begin{aligned}
\widetilde{f}'_j(\boldsymbol{\theta}_j) &= \prod_{\theta_l \in \boldsymbol{\theta}_j} \widetilde{f}'_{jl}(\theta_l) \\
&\propto \frac{q^*(\boldsymbol{\theta})}{q^{\backslash j}(\boldsymbol{\theta})} = \frac{q^{\backslash j}(\boldsymbol{\theta}) \prod_{\theta_l \in \boldsymbol{\theta}_j} \left(\int f_j(\boldsymbol{\theta}_j) \prod_{n\neq j, \theta_k \in \boldsymbol{\theta}_j \backslash \theta_l} \widetilde{f}_{nk}(\theta_k) \mathrm{d}\{\boldsymbol{\theta}_j \backslash \theta_l\} \right)}{q^{\backslash j}(\boldsymbol{\theta})} \\
&= \prod_{\theta_l \in \boldsymbol{\theta}_j} \left(\int f_j(\boldsymbol{\theta}_j) \prod_{n\neq j, \theta_k \in \boldsymbol{\theta}_j \backslash \theta_l} \widetilde{f}_{nk}(\theta_k) \mathrm{d}\{\boldsymbol{\theta}_j \backslash \theta_l\} \right)
\end{aligned}
$$

可得 factor $\widetilde{f}'_j(\boldsymbol{\theta}_j)$ 中子 factor $\widetilde{f}'_{jl}(\theta_l)$ 的计算公式（对应书中的式 (10.240)，但这里表述得更清晰）：

$$
\widetilde{f}'_{jl}(\theta_l) \propto \int f_j(\boldsymbol{\theta}_j) \prod_{n\neq j, \theta_k \in \boldsymbol{\theta}_j \backslash \theta_l} \widetilde{f}_{nk}(\theta_k) \mathrm{d}\{\boldsymbol{\theta}_j \backslash \theta_l\}, \text{ for all } \theta_l \in \boldsymbol{\theta}_j
$$

上面的推导只看公式可能不是很直观，但画个概率图作为辅助，就很清晰了。

按 EP 算法的步骤继续走，令 $K_j = Z_j$，进而也确定了 $\widetilde{f}'_j(\boldsymbol{\theta}_j)$ 的大小，再令 $\widetilde{f}_j(\boldsymbol{\theta}_j) \leftarrow \widetilde{f}'_j(\boldsymbol{\theta}_j)$，至此，更新了一次 factor $\widetilde{f}_j(\boldsymbol{\theta}_j)$。接着，选择下一个 factor 继续更新下去。注意，对子 factor $\widetilde{f}'_{jl}(\theta_l)$，我们并不能确定它们的具体大小，只要所有子 factor 累乘后大小为 $\widetilde{f}'_j(\boldsymbol{\theta}_j)$ 即可。

从 EP 到 BP：message passing

下面，我们对上述 EP 算法进行修改，得到相应的 message passing 算法。事实上，为了得到 message passing 算法，对 EP 的修改还是挺大的，都能算得上是面目全非了，不过我们还是尊重书本，认为后者是前者的特例。

> 我们首先对几个概念进行梳理和说明：
>
> - belief propagation (BP) 算法全书并未详细给出，只是在 P403 中提到 BP 是 sum-product 的特例。根据书中语义，我们可以认为提到 BP 算法时就是在说 sum-product 算法。BP 算法对应 tree-structure 的概率图，是一种精确推断算法，只需 2 轮信息传递。书中 8.4.7 小节介绍了 loopy BP 算法，对应存在环的概率图模型，因此是一种近似推断算法，需要不断迭代更新，而且不保证收敛。

> - (loopy) BP 在求积分，而 EP 在求近似分布，表面上看二者似乎并不相关，但之所以要求近似分布，是因为我们虽然知道分布的函数形式，但却不知道归一化常量，而归一化常量的计算就是一个积分问题；从另一角度，(loopy) BP 求积分的应用之一正是计算归一化常量以获得完整的概率分布。因此，在某种程度上二者都是在做同一件事，目标相同，所以我们才有可能对 EP 进行修改得到 (loopy) BP，说后者是前者的特例。

我们指出，能将 EP 修改为 message passing 算法的核心在于子 factor $\widetilde{f}_{jl}(\theta_l)$ 的更新公式 (和前面一样，为了表述更清晰，对更新的量，我们加上上标 $'$ 以示区分)：

$$\widetilde{f}'_{jl}(\theta_l) \propto \int f_j(\boldsymbol{\theta}_j) \prod_{n\neq j, \theta_k\in\boldsymbol{\theta}_j\backslash\theta_l} \widetilde{f}_{nk}(\theta_k)\mathrm{d}\{\boldsymbol{\theta}_j\backslash\theta_l\},\ \text{for all}\ \theta_l\in\boldsymbol{\theta}_j$$

与 message passing 的公式在形式上相同（虽然前者是 $\propto$ 而后者是 $=$，但这并没有太大影响，我们可以令 $\propto$ 取 $=$，虽然这隐式地确定了 K_j 与 $\widetilde{f}_j(\boldsymbol{\theta}_j)$ 间的分配，可能会使 $K_j\neq Z_j$ 而与前面标准 EP 算法不同，但这并没有关系，因为从后面我们可以看到，为了得到 BP，修改后的 EP 压根就没按照 $p(\mathcal{D})\approx\int\prod_i\widetilde{f}_i(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta},\ p(\boldsymbol{\theta})\approx q(\boldsymbol{\theta})=\prod_i\widetilde{f}_i(\boldsymbol{\theta}_i)/\int\prod_i\widetilde{f}_i(\boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta}$ 来计算最终结果）：

$$\mu_{j\to\theta_l}(\theta_l)=\int f_j(\boldsymbol{\theta}_j)\prod_{n\neq j,\theta_k\in\boldsymbol{\theta}_j\backslash\theta_l}\mu_{n\to\theta_k}(\theta_k)\mathrm{d}\{\boldsymbol{\theta}_j\backslash\theta_l\}$$

从内容和形式的辩证关系出发，我们可以将子 factor $\widetilde{f}_{jl}(\theta_l)$ / $\widetilde{f}'_{jl}(\theta_l)$ 赋予从因子结点 j 到变量结点 θ_j 的信息的含义 (这里不显示地涉及变量结点到因子结点这种类型的信息)，从而启发我们是否可以对 EP 作一定的修改得到 message passing 算法（虽然个人认为都不能叫修改，还不如直接说是因为 EP 能推得上述公式，而且如前文所述，EP 和 BP 在某种程度上都是在做同一件事，基于这两点，我们就强行认为 BP 是 EP 的特例。不过，下面我们还是假装可以通过“修改” EP 得到 BP）。

为了推得上述和 (loopy) BP 相同的计算公式，我们需要 remove 整个 factor $\widetilde{f}_j(\boldsymbol{\theta}_j)=\prod_{\theta_l\in\boldsymbol{\theta}_j}\widetilde{f}_{jl}(\theta_l)$ 即 $q^{\backslash j}$。标准 EP 会更新整个 factor $\widetilde{f}_j(\boldsymbol{\theta}_j)$ 中的每个子 factor，但为了推得 (loopy) BP，我们需要按照 (loopy) BP 信息传递的顺序来更新。注意，并不是像书中说得每次只更新一个子 factor。比如，对 tree-structure 的概率图模型，即 BP 算法，第一轮更新时（传递方向由叶子结点向 root 结点），只会更新因子结点向 root 结点方向的信息（子 factor），而该因子结点向其余方向的信息 (子 factors) 则可以在第二轮（传递方向由 root 向叶子结点）同时更新，它们的计算相互之间不受影响（这里可以画个图方便理解。另外，再次强调，这里不显示地涉及 variable node to factor node message，子 factor 对应的是 factor node to variable node message）。

在同步了计算顺序后，还有一个问题就是初始化。可以看到，虽然 EP 需要初始化近似分布 $q(\boldsymbol{\theta})$，(loopy) BP 则压根就不需要引入近似分布，全程基于 $p(\boldsymbol{\theta})$ 进行计算，但事实上，BP 语义中的 message 就对应近似分布，可以看到：

- 概率图 tree-structure 时，因为 BP 特定的更新顺序，对 $q(\boldsymbol{\theta})$（message）如何初始化都不会影响计算结果（按传递顺序走一遍即可证明这一点），且两轮更新就会收敛；
- 概率图存在环时，loopy BP 需要初始化 message：

  > We have seen that a message can only be sent across a link from a node when all other messages have been received by that node across its other links. Because there are loops in the graph, this raises the problem of how to initiate the message passing algorithm. To resolve this, we suppose that an initial message given by the unit function has been passed across every link in each direction. Every node is then in a position to send a message. ——《PRML》P417

  因此，我们也需要将所有的子 factor 初始化为 1.

在令子 factor 更新顺序与 message 传递顺序保持一致，并且初始化也相同后，为了得到 BP，在计算最终结果时 EP 还需要放弃标准的计算近似分布 $q(\boldsymbol{\theta}) = \prod_i \widetilde{f}_i(\boldsymbol{\theta}_i)/Z = \prod_i \prod_{\theta_k \in \boldsymbol{\theta}_i} \widetilde{f}_{ik}(\theta_k)/Z$, $Z = \int \prod_i \widetilde{f}_i(\boldsymbol{\theta}_i) \mathrm{d}\boldsymbol{\theta}$，并用其替代目标分布 $p(\boldsymbol{\theta})$ 的方案。不再将子 factor $\widetilde{f}_{ik}(\theta_k)$ 作为用来近似原 factor $f_i(\boldsymbol{\theta}_i)$ 的 $\widetilde{f}_i(\boldsymbol{\theta}_i)$ 的组成部分使用，而是完全像 BP 那样以 message 的方式对待子 factor，即子 factor 是因子结点传递给变量结点的信息，表示概率图相应部分的积分，我们将某个变量结点收到的所有因子结点的信息累乘，再对这个变量积分，即可得归一化常量，从而得到完整的概率分布 $p(\boldsymbol{\theta})$。以 tree-structure 时的 BP 为例，BP 得到的是 $p(\boldsymbol{\theta})$ 的精确解，而按标准 EP 走完，得到的依然是近似解。这也很好理解，毕竟 $p(\boldsymbol{\theta})$ 可能压根不在 $q(\boldsymbol{\theta}) \propto \prod_i \widetilde{f}_i(\boldsymbol{\theta}_i) = \prod_i \prod_{\theta_k \in \boldsymbol{\theta}_i} \widetilde{f}_{ik}(\theta_k)$ 的函数空间中。

总而言之，在更新顺序一致 + 初始化相同 + 放弃 EP 出结果的标准方案而完全采用 BP 的方案后，我们终于从 EP "推得" 了 BP。可以看到，EP 实际上就只出了一个更新公式，而其它过程都要强行变得和 BP 相同才能得到 BP，因此说 BP 是 EP 的特例实属有点牵强附会了。

最后：

1. VI/VB: $\min_q \mathrm{KL}(q\|p)$. 进一步，若 q 的 factors depend only on subsets of the variables，即应用在概率图模型上时，可推得 variational message passing (书中 10.4.1 小节，但没有详谈)；
2. EP: $\min_q \mathrm{KL}(p\|q)$. 同样地，当应用在概率图模型上时，即得 (loopy) BP 算法（虽然很牵强）。此外，虽然 EP 名字含 propagation，但标准的 EP 没有"传递"什么东西；
3. 更一般地，有最小化 the alpha family of divergences，即 $\min_q \mathrm{D}_\alpha(p\|q)$，当应用在概率图模型上时，也存在相应的 message passing 算法。
