test

17 June, 2018

目录

MLAPP 读书笔记

21 变分推断法 (Variational Inference)

A Chinese Notes of MLAPP，MLAPP 中文笔记项目 https://zhuanlan.zhihu.com/python-kivy

记笔记的人：shouldsee

21.1 导论

我们在本书中已经探讨了好几种计算后验分布（以及其函数）的算法。对于离散的图模型，我们可以用链接树算法（JTA:junction tree algorithm）来进行确切 (exact) 的推断，前文**??**已经加以讨论。但是，这个算法的时间复杂是图链接树的宽度的指数函数，由此导致确切推断常常是不现实的。对于高斯图模型，确切推断的时间复杂度是树宽的三次函数。然而，即便这个算法在我们有很多变量的时候也会变得奇慢无比。另外，链接树算法对非高斯的连续变量以及离散/连续相混合的变量是束手无策的。

对于某些简单共用 $x->D$ 形式的的双节点图模型, 我们可以在先验分布 $p(x)$ 和似然函数共轭的情况下，计算其后验分布 $p(x|D)$ 确切的闭式解（也就是说似然函数必须是一个指数族分布），更多例子见 Chap5**??**（注意在本章节中 x 代表未知变量，而在 Chap5**??**中我们用的是 θ 表示未知数）。

在更一般的情况下，我们必须使用近似的推断法。在 Sec8.4.1**??**中，我们讨论了高斯近似在非共轭的双节点中的应用（如 Sec8.4.3**??**中在逻辑回归上的应用。）

高斯近似是简单的。但是，有一些后验分布不能用高斯分布很好地模拟。比如说，在推断多项分布的参数时，狄利克雷分布是一个更好的选择；推断离散图模型的状态变量时，分类分布是一个更好的选项。

在本章节中，我们考虑一个更加一般的、基于变分推断的确定性近似推断算法（**???**，**???**,**???**，**???**）。其基本理念是从易处理的分布族中选取

一个近似分布 $q(x)$，然后设法让这个近似分布尽可能地接近真正的后验 $p^*(x) \triangleq p(x|D)$。这样以来原本的推断问题就简化成了一个最优化问题。通过放宽限制或者对目标函数再次作近似，我们可以在精度和速度之间寻找平衡。因此变分推断可以用最大后验估计（MAP）算法的速度实现贝叶斯方法的统计学优势。

在本章节中，我们考虑一个更加一般的、基于变分推断的确定性近似推断算法 (**???**，**???**,**???**，**???**)。其基本理念是从易处理的分布族中选取一个近似分布 $q(x)$，然后设法让这个近似分布尽可能地接近真正的后验 $p^*(x) \triangleq p(x|D)$。这样以来原本的推断问题就简化成了一个最优化问题。通过放宽限制或者对目标函数再次作近似，我们可以在精度和速度之间寻找平衡。因此变分推断可以用最大后验估计（MAP）算法的速度实现贝叶斯方法的统计学优势。

21.2 变分推断法

假设 $p^*(x)$ 使我们的真实却难以处理的分布，而 $q(x)$ 是某个便于处理的近似分布，比如说一个多维高斯分布或者因子分解过的分布。我们假设 q 具有一些可以自由参数，并且我们可以通过优化这些参数使得 q 更加像 p^*。我们显然可以最小化损失函数 KL 散度:

$$\mathbb{KL})(p^*||q) = \sum_x p^* x \log \frac{p^*(x)}{q(x)} \quad (21.1)$$

但是，这玩意非常难算，因为在分布 p^* 上求期望根据题设是难以处理的。一个自然的替代选项是最小化逆 KL 散度:

$$\mathbb{KL})(q||p^*) = \sum_x q(x) \log \frac{q(x)}{p^*(x)} \quad (21.2)$$

这个目标函数最大的优势是在分布 q 上的期望是便于计算的 (通过选取适当形式的 q)。我们会在 Sec21.2.2**??**中讨论这两个目标函数的区别。

不幸的是，公式 Eqn21.2 仍然没有看起来那么好算，因为即便逐点计算 $p(x|D)$ 也是很困难的，因为有一个正规化常数 $Z = p(D)$ 是难以处理的。但是呢，一般来说未正规化的分布 $\tilde{p}(x) \triangleq p(x, D) = p^*(x)|$ 是很好计算的。所以我们将目标函数改为如下:

$$Jq = \mathbb{KL}(q||\tilde{p}) \quad (21.3)$$

当然这个写法有点滥用记号的意思，因为 $\tilde{p}$ 严格意义上讲并不是一个概率分布。不过无所谓，让我们带入 KL 散度的定义：

$$J(q) = \sum_x q(x) \log \frac{q(x)}{\tilde{p}(x)} \quad (21.4) \tag{1}$$

$$= \sum_x q(x) \log \frac{q(x)}{Zp^*(x)} \quad (21.5) \tag{2}$$

$$= \sum_x q(x) \log \frac{q(x)}{p^*(x)} - \log Z \quad (21.6) \tag{3}$$

$$= \mathbb{KL}(q||p^*) - \log Z \quad (21.7) \tag{4}$$

因为 Z 是一个常数，所以最小化 $J(q)$ 的同时我们也就达到了迫使 q 趋近 p^* 的目的。

因为 KL 散度总是非负的，可以看出 $J(q)$ 是负对数似然 (NLL:Negative Log Likelihood) 的上界:

$$J(q) = \mathbb{KL}(q||p^*) - \log Z \geq -\log Z = -\log pD \quad (21.8)$$

换句话说，我们可以尝试最大化如下被称作能量泛函的量 (**???**)。它同时也是数据似然度的下界：

$$L(q) \triangleq -J(q) = -\mathbb{KL}(q||p^*) + \log Z \leq \log Z = \log p(D) \quad (21.9)$$

因为这个界在 $q = p^*$ 时是紧的，可以看出变分推断法和 EM 算法联系之紧密 (见 Sec11.4.7**??**)。

21.2.1 变分目标函数的其他意义

前述的目标函数还有几种同样深刻的写法。其中一种如下：

$$J(q) = \mathbb{E}_q[\log q(x)] + \mathbb{E}[-\log \tilde{p}(x)] = -\mathbb{H}(q) + \mathbb{E}_q[E(x)] \quad (21.10)$$

也就是能量的期望（因为 $E(x) = -\log \tilde{p}(x)$）减去系统的熵。在统计物理里，$J(q)$ 被称为变分自由能，或者也叫亥姆霍兹自由能。

另一写法如下:

$$J(q) = \mathbb{E}_q[\log q(x) - \log p(x)p(D|x)] \quad (21.11) \tag{5}$$

$$= \mathbb{E}_q[\log q(x) - \log p(x) - \log p(D|x)] \quad (21.12) \tag{6}$$

$$= \mathbb{E}_q[-\log p(D|x)] + \mathbb{KL}(q(x)||p(x)) \quad (21.13) \tag{7}$$

也就是负对数似然的期望加上一个表示后验分布到确切的先验距离的惩罚项。

我们还可以从信息论的角度理解（也叫作bits-back 论述，具体见(???,???)）。

21.2.2 前向 (Forward) 还是后向 (Reverse)KL 散度？

因为 KL 散度是非对称的，对 q 最小化 $\mathbb{KL}(q||p)$ 和 $\mathbb{KL}(p||q)$ 会给出不同的结果。接下来我们讨论一下两者的异同。

首先考虑后向 KL，$\mathbb{KL}(q||p)$，也称 **I-投影**或**信息投影**。根据定义有

$$\mathbb{KL}(q||p) = \sum_x q(x) \ln \frac{q(x)}{p(x)} \quad (21.14)$$

这个量在 $p(x) = 0$ 且 $q(x) > 0$ 是无穷的。因此如果 $p(x) = 0$ 就必须要有 $q(x) = 0$。因此后向 KL 被称作是**迫零的**，而且近似分布 q 常常会欠估计 p 的支撑集。

接下来考虑前向 KL, 也称 **M-投影**或者**矩投影**

$$\mathbb{KL}(p||q) = \sum_x p(x) \ln \frac{p(x)}{q(x)} \quad (21.15)$$

这个量在 $q(x) = 0$ 且 $p(x) > 0$ 是无穷的。因此如果 $p(x) > 0$ 就必须要有 $q(x) > 0$。因此前向 KL 被称作**避零的**，而且近似分布 p 常常会过估计 p 的支撑集。

两者的区别请见图 Fig21.1。可以发现当真实分布 p 是多模态的时候，前向 KL 是一个很差的选择 (此处假设使用了一个单模态的 q), 因为它给出的后验的众数/平均数会落在一个低密度的区域，恰恰落在两个模态的峰值之间。在这个情况下，后向 KL 不仅更便于计算，也具有更好的统计性质。

另一个区别显示在图 Fig 中。此处的真实分布是一个拉长的二维高斯分布而近似分布是两个一维高斯的乘积。换句话说 $p(x) = \mathcal{N}(x|\mu, \Lambda^{-1})$，且

$$\mu = \begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \Lambda = \begin{pmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{pmatrix}$$

在图 Fig21.2a**??**中我们给出了最小化后向 KL $\mathbb{KL}(q||p)$ 的结果。在这个例子中，我们可以证明有如下解

$$q(x) = \mathcal{N}(x_1|m_1, \Lambda_{11}^{-1})\mathcal{N}(x_2|m_2, \Lambda_{22}^{-1}) \quad (21.17) \tag{8}$$

$$m1 = \mu_1 - \Lambda_{11}^{-1}\Lambda_{12}(m_2 - \mu_2) \quad (21.18) \tag{9}$$

$$m2 = \mu_2 - \Lambda_{22}^{-1}\Lambda_{21}(m_1 - \mu_1) \quad (21.19) \tag{10}$$

图 Fig21.2a**??**的结果显示我们正确地估计了平均值，但是这个近似太紧凑了：它的方差是由 p 的方差最小的那个方向所决定的。事实上，通常情况下（当然也有特例 **???**）在 q 是乘积分布时最小化 $\mathbb{KL}(q||p)$ 会给出一个过于置信的近似。

图 Fig21.2b**??**给出了最小化 $\mathbb{KL}(p||q)$ 的结果。在习题 Exer21.7**??**我们已经证明对一个乘积分布最小化正向 KL 给出的最优解正好是其真实边际分布的乘积，也就是说有

$$q(x) = \mathcal{N}(x_1|\mu_1, \Lambda_{11}^{-1})\mathcal{N}(x_2|\mu_2, \Lambda_{22}^{-1}) \quad (21.20)$$

图 Fig21.2b**??**显示出这个估计是过泛的，因为它过估计了 p 的支撑集。

在本章的剩余部分，以及本书接下来的大部，我们会专注于最小化后向 KL $\mathbb{KL}(q||p)$。在 Sec22.5**??**对期望传播的阐述中，我们会探讨前向 KL $\mathbb{KL}p||q$ 在局部的优化。

21.2.3 另外的一些相关的度量

通过引入参数 $\alpha \in \mathbb{R}$ 我们可以定义如下 **alpha 散度**：

$$D_\alpha(p||q) \triangleq \frac{4}{1-\alpha^2}\left(1 - \int p(x)^{(1+\alpha)/2} q(x)^{(1-\alpha)/2} dx\right) \quad (21.21)$$

这个量满足 $D_\alpha(p||q) \iff p = q$，但是它们显然也是不对称的，因而不是一个度规。$\mathbb{KL}(p||q)$ 对应极限 $\alpha \to 1$ ，而 $\mathbb{KL}(q||p)$ 对应极限 $\alpha \to -1$。当 $\alpha = 0$，我们取得一个和海灵格距离线性相关的对称的散度，定义如下

$$D_H(p||q) \triangleq \int \left(p(x)^{\frac{1}{2}} - q(x)^{\frac{1}{2}}\right)^2 \quad (21.22)$$

注意到 $\sqrt{D_H(p||q)}$ 是一个有效的距离度规。也就是说，它对称非负且满足三角不等式，详见 (**???**)。

21.5 变分贝叶斯（变贝）

目前为止我们主要专注于在模型参数 θ 已知的情况下推断隐变量 z_i 的分布。现在起我们开始考虑推断模型参数其本身。如果我们作一个完全的因子分解（也就是平均场）近似 $p(\theta|D) \approx \prod_k q(\theta_k)$，我们实际上就在做**变分贝叶斯** (VB:Variational Bayes, **???**,**???**,**???**,**???**,**???**) 了。接下来我们会给出一些在假设不存在隐变量情况下，应用变分贝叶斯的例子，如果我们想要同时推断隐变量和参变量，并作如下近似 $p(\theta, z_{1:N}|D) \approx q(\theta)\prod_i q_i(z_i)$，我们实际上就在使用变分贝叶斯期望最大化算法 (VBEM)，详见 Sec21.6**??**。

21.5.1 实例：单一高斯变量的变贝

根据 (**???**,p429)，我们考虑推断一维高斯分布的参数分布 $p(\mu, \lambda|D)$，此处 $\lambda = 1/\sigma^2$ 指高斯精度。为简便期间，我们使用一个如下的共轭先验分布

$$p(\mu, \lambda) = \mathcal{N}(\mu|\mu_0, (\kappa\lambda)^{-1})\mathrm{Ga}(\lambda|a_0, b_0) \quad (21.65)$$

但是呢，我们用来近似后验的是一个如下的因子分布

$$q(\mu, \lambda) = q_\mu(\mu)q_\lambda(\lambda) \quad (21.66)$$

我们不需要指定分布 q_μ, q_λ 的确切形式，因为其最优形式会在推导的时候自动 “出现”(方便的是，它们恰巧是相应的高斯分布和伽马分布)

由于我们在 Sec4.6.3.7**??**已经得出了计算该模型确切后验的方法，因此如上处理的动机并不是很显然。动机主要有如下两条：首先这是一个有教学意义的练习，我们可以乘机通过与确切分布的对比对比，来分析近似分布的质量；其次，这个方法可以很自然地加以改造以适用一个半共轭的先验 $p(\mu, \lambda) = \mathcal{N}(\mu|\mu_0, \tau_0)\mathrm{Ga}(\lambda|a_0, b_0)$，而这种先验下是无法进行确切推断的。

21.5.1.1 目标分布

未正规化的对数后验具有形式

$$\log \tilde{p}(\mu,\lambda) = \log p(\mu,\lambda,D) = \log p(D|\mu,\lambda) + \log p(\mu|\lambda) + \log p(\lambda) \quad (21.67)$$

(11)

$$= \frac{N}{2}\log\lambda - \frac{\lambda}{2}\sum_{i=1}^{N}(x_i-\mu)^2 - \frac{\kappa_0\lambda}{2}(\mu-\mu_0)^2$$

(12)

$$+ \frac{1}{2}\log(\kappa_0\lambda) + (a_0-1)\log\lambda - b_0\lambda + \text{const} \quad (21.68)$$

(13)

21.5.1.2 更新 $q_\mu\mu$

通过对 λ 求期望可以得到 $q_\mu(\mu)$ 的最优形式:

$$\log q_\mu(\mu) = \mathbb{E}_{q_\lambda}[\log p(D|\mu,\lambda) + \log p(\mu|\lambda)] + \text{const} \quad (21.69) \quad (14)$$

$$= -\frac{\mathbb{E}[\lambda]}{2}\left\{\kappa_0(\mu-\mu_0)^2 + \sum_{i=1}^{N}(x_i-\mu)^2\right\} + \text{const} \quad (21.70) \quad (15)$$

通过展开平方项 (? 疑) 可以证明 $q_\mu(\mu) = (\mu|\mu_N, \kappa_N^{-1})$, 且有

$$\mu_N = \frac{\kappa_0\mu_0 + N\bar{x}}{\kappa_0+N}, \kappa_N = (\kappa_0+N)\mathbb{E}_{q_\lambda}[\lambda] \quad (21.71)$$

此时我们还不知道 $q_\lambda(\lambda)$ 的确切分布所以无法计算 $\mathbb{E}[\lambda]$, 但我们马上就会解决这个玩意

21.5.1.3 更新 $q_\lambda(\lambda)$

$q_\lambda(\lambda)$ 的最优分布有如下表示

$$\log q_\lambda(\lambda) = \mathbb{E}_{q_\mu}[\log p(D|\mu,\lambda) + \log p(\mu|\lambda) + \log p(\lambda)] + \text{const} \quad (21.72)$$

(16)

$$= (a_0 - 1)\log\lambda - b_0\lambda + \frac{1}{2}\log\lambda + \frac{N}{2}\log\lambda \quad (17)$$

$$- \frac{\lambda}{2}\mathbb{E}_{q_\mu}\left[\kappa_0(\mu-\mu_0)^2 + \sum_{i=1}^{N}(x_i-\mu)^2\right] + \text{const} \quad (21.73) \quad (18)$$

可以发现这对应一个伽马分布，因而有 $q_\lambda(\lambda) = Ga(\lambda|a_N, b_N)$, 且有

$$a_N = a_0 + \frac{N+1}{2} \quad (21.74) \quad (19)$$

$$b_N = b_0 + \frac{1}{2}\mathbb{E}_{q_\mu}\left[\kappa_0(\mu-\mu_0)^2 + \sum_{i=1}^{N}(x_i-\mu)^2\right] \quad (21.75) \quad (20)$$

21.5.1.4 计算期望

为了实行上述更新，我们必须确定如何计算诸多期望值。鉴于 $q(\mu) = (\mu|\mu_N, \kappa_N^{-1})$, 我们得出

$$\mathbb{E}_{q(\mu)}[\mu] = \mu_N \quad (21.76) \quad (21)$$

$$\mathbb{E}_{q(\mu)}[\mu^2] = \frac{1}{\kappa_N} + \mu_N^2 \quad (21.77) \quad (22)$$

鉴于 $q(\lambda) = \text{Ga}(\lambda \mid a_N, b_N)$，我们有

$$\mathbb{E}_{q(\lambda)}[\lambda] = \frac{a_N}{b_N} \quad (21.78)$$

如此我们可以显式地给出更新方程。对 $q(\mu)$ 有

$$\mu_N = \frac{\kappa_0\mu_0 + N\bar{x}}{\kappa_0 + N} \quad (21.79) \quad (23)$$

$$\kappa_N = (\kappa_0 + N)\frac{a_N}{b_N} \quad (21.80) \quad (24)$$

相应地对 $q(\lambda)$ 有

$$a_N = a_0 + \frac{N+1}{2} \quad (21.81) \tag{25}$$

$$b_N = b_0 + \kappa_0(\mathbb{E}[\mu^2] + \mu_0^2 - 2\mathbb{E}[\mu]\mu_0) + \frac{1}{2}\sum_{i=1}^{N}(x_i^2 + \mathbb{E}[\mu^2] - 2\mathbb{E}[\mu]x_i) \quad (21.82)$$

$$\tag{26}$$

可以看到 μ_N 和 a_N 实际上是固定的常数，且只有 κ_N ,b_N 需要被迭代更新。(事实上，我们可以利用迭代方程，解析地求出 κ_N b_N 的不动点，但是在此不作展示，只介绍迭代更新法)

21.5.1.5 展示

配图 Fig21.5**??**给出了一个工作示意。绿色的轮廓表示确切的高斯-伽马后验分布，虚线的轮廓表示的是变分近似在不同的迭代步的结果。可以看到最终的近似是和确切解较为相似的。然而，近似解比真实分布更为“紧致”。事实上平均场近似推断常常欠估计后验的不确定性，更多有关讨论请见 Sec21.2.2**??**。

21.5.1.6 下界

在变贝中，我们最大化的是 $L(q)$ 是数据的边际似然值的下界。

$$L(q) \leq \log p(D) = \log \int\int p(D\mu)p(\mu,\lambda)d\mu d\lambda \quad (21.83)$$

出于多种原因，计算这个下界本身是很有用的。首先，下界可以用来检查算法是否收敛了；其次，它可以用来检查算法的正确性：和期望最大算法 (EM) 的情况一样，如果下界不是单调增加的话，那一定是哪里出幺蛾子了；再次，下界可以用作一个对数据边际似然的近似，因而可以用来做贝叶斯模型选择。

不幸的是，计算这个下界需要经过一些很是心累的代数处理。以下我们会给出这个特例的下界的细节，但我们将不经证明，甚至不加讨论地直接给出下界的结果，因为这样子会简洁许多（? 我的天）。

对这个模型，我们可以如下计算似然函数 $L(q)$

$$L(q) = \int\int q(\mu,\lambda)\log\frac{p(D,\mu,\lambda)}{q(\mu,\lambda)}d\mu d\lambda \quad (21.84) \tag{27}$$

$$= \mathbb{E}[\log p(D\ gvn\mu,\lambda)] + \mathbb{E}[\log p(\mu\mid\lambda)] + \mathbb{E}[\log p(\lambda)] \tag{28}$$

$$- \mathbb{E}[\log q(u)] - \mathbb{E}[\log q(\lambda)] \quad (21.85) \tag{29}$$

以上所有的期望都是关于 $q(\mu,\lambda)$。可以看出最后两项不过是高斯和伽马分布的熵，因此可以直接写出

$$\mathbb{H}(\mathcal{N}(\mu_N,\kappa_N^{-1}) = -\frac{1}{2}\log\kappa_N + \frac{1}{2}(1+\log(2\pi)) \quad (21.86) \tag{30}$$

$$\mathbb{H}(\mathrm{Ga}(a_N,b_N)) = \log\Gamma(a_N) - (a_N-1)\psi(a_N) - \log(b_N) + a_N \quad (21.87) \tag{31}$$

此处 $\psi()$ 是双伽马函数。

为了计算其它的项目，我们要用到如下结果

$$\mathbb{E}[\log x\mid x\sim\mathrm{Ga}(a,b)] = \psi(a) - \log(b) \quad (21.88) \tag{32}$$

$$\mathbb{E}[x\mid x\sim\mathrm{Ga}(a,b)] = \frac{a}{b} \quad (21.89) \tag{33}$$

$$\mathbb{E}[x\mid x\sim\mathcal{N}(\mu,\sigma^2)] = \mu \quad (21.90) \tag{34}$$

$$\mathbb{E}[x^2\mid x\sim\mathcal{N}(\mu,\sigma^2)] = \mu^2+\sigma^2 \quad (21.91) \tag{35}$$

可以证明对数似然的期望符合下式

$$\mathbb{E}_{q(\mu,\lambda)}[\log p(D\mid\mu\lambda)] \quad (21.92) \tag{36}$$

$$= -\frac{N}{2}\log(2\pi) + \frac{N}{2}\mathbb{E}_{q_\lambda}[\log\lambda] - \frac{\mathbb{E}_{q_\lambda}[\lambda]}{2}\sum_{i=1}^{N}\mathbb{E}_{q(\mu)}[(x_i-\mu)^2] \quad (21.93) \tag{37}$$

$$= -\frac{N}{2}\log(2\pi) + \frac{N}{2}(\psi(a_N) - \log b_N) \tag{38}$$

$$- \frac{Na_N}{2b_N}\left(\hat{\sigma}^2 + \bar{x}^2 - 2\mu_N\bar{x} + \mu_N^2 + \frac{1}{\kappa_N}\right) \quad (21.94) \tag{39}$$

此处的 $\bar{x}$, $\hat{\sigma}^2$ 对应观测平均和方差。

对 λ 的对数先验的期望可做如下处理

$$\mathbb{E}_{q(\lambda)}[\log p(\lambda)] = (a_0-1)\mathbb{E}[\log\lambda] - b_0\mathbb{E}[\lambda] + a_0\log b_0 - \log\Gamma(a_0) \quad (21.95)$$

(40)

$$= (a_0-1)(\psi(a_N) - \log b_N) - b_0\frac{a_N}{b_N} + a_0\log b_0 - \log\Gamma(a_0) \quad (21.96)$$

(41)

对 μ 的对数先验的期望可做如下处理

$$\mathbb{E}_{q(\mu,\lambda)}[\log p(\mu \mid \lambda)] = \frac{1}{2}\log\frac{\kappa_0}{2\pi} + \frac{1}{2}\mathbb{E}[\log\lambda]q(\lambda) - \frac{1}{2}\mathbb{E}_{q(\mu,\lambda)}[(\mu-\mu_0)^2\kappa_0\lambda]$$

(42)

$$= \frac{1}{2}\log\frac{\kappa_0}{2\pi} + \frac{1}{2}(\psi(a_N) - \log b_N) \quad (43)$$

$$-\frac{\kappa_0}{2}\frac{a_N}{b_N}\left[(\mu_N-\mu_0)^2 + \frac{1}{\kappa_N}\right] \quad (21.97) \quad (44)$$

将以上结果结合，可以得到对数似然的下界

$$L(q) = \frac{1}{2}\log\frac{1}{\kappa_N} + \log\Gamma(a_N) - a_N\log b_N + \text{const} \quad (21.98)$$

这个量随着变贝的迭代更新是单调递增的。

21.5.2 实例：线性回归中的变贝

（？略略略）

21.6 变分贝叶斯期望最大算法 (变贝期最,VBEM：Variational Bayes Expectation Maximisation)

现在让我们考虑具有形式 $z_i \to x_i \leftarrow \theta$。这囊括了混合模型，主成分分析，隐马尔可夫模型，等等。未知的变量现在有两种: 参变量 θ，以及隐变量 z_i。在章节 Sec11.4**??**我们讨论过这类模型而且它们通常是用 “期望-最大化” 算法来拟合的。算法的 “期望” 步 (E-step) 会推断一个关于隐变量的后验分布 $p(z_i \mid x_i, \theta)$；在 “最大化” 步 (M-step)，我们计算一个关于参变量 θ

的点估计。作此处理的原因有两个：首先，这会给出一个简单的算法；其次，参变量 θ 的后验不定性通常小于隐变量 z_i 的，鉴于 θ 从所有 N 个样本中得到信息，而 z_i 只从 x_i 获得信息。这就使得对 θ 而不是 z_i 作点估计显得更为合理。

不过呢，变贝通过同时模拟参变量 θ 和隐变量 z_i 的不定性给出了一个更加“贝叶斯化”的算法，同时又和期望最大算法有着差不多的计算速度。这个算法因此叫做变贝期最 (VBEM)。其基本理念是用平均场来近似后验分布

$$p(\theta, z_{1:N} \mid D) \approx q(\theta)q(z) = q(\theta)\prod_i q(z_i) \quad (21.120)$$

第一个关于 θ, z_i 的因子分解是一个简化算法的关键假设。第二个因子分解可以从模型对隐变量在给定参变量 θ 后独立同分布的的假设中得出。

在变贝期最中，我们交替更新 $q(z_i|D)$ (变分“期望”步)，和更新 $q(\theta|D)$ (变分”最大化“步)。要从变贝期最可以回到期最算法，只需用狄拉克函数来表示对 θ 的点估计也即 $q(\theta \mid D) \approx \delta_{\hat{\theta}}(\theta)$

变分期望步和正常期望步是差不多的，不过变分期望步并不计算基于点估计的后验分布 $p(z_i \mid D, \hat{\theta})$，而是需要对参数分布求期望/边际分布。粗略来讲。我们可以用参变量后验的平均值来代替点估计来得到隐变量的后验分布 $p(z_i \mid D, \hat{\theta})$，然后直接扔上经典如” 前向-后向“算法。不幸的是，事情并不总是那么简单，不过基本思想都和这是一样的。更多细节常常要根据模型的形式决定，我们接下来会通过实例说明。

变分的最大化步和正常的最大化步是相似的，不过此时我们更新的是参变量分布的超参数而不是它的点估计，这是借由充分统计量的期望实现的。这个步骤通常和正常的点估计很相似，但是呢细节还是要等到模型定下来才能讨论。

变贝期最相对于经典期最的主要优势在于我们可以通过边际化参变量分布来计算数据的边际似然值，然后用这个似然值去做模型选择。我们会在 Sec21.6.1.6**??**看到一个实例。同时呢，变贝期最还是奉行“平等主义”的，因为它并不把参变量视为“二等公民”，而是和其他未知量平等地联立；相比之下，期最算法人为地区分了参变量和隐变量。

21.6.1 实例：高斯混合分布的变贝期最

现在让我们考虑用变贝期最来“拟合”一个高斯混合模型 (此处的引号是因为我们并不是在估计模型参数，而是在推断它们的后验分布)。我们会基于 (**???**,Sec 10.2) 阐述。不幸的是，这些细节都十分的繁杂。不过万幸的是，同期最算法一样，算多了你就自然而然知道该怎么搞了 (?-_-),(和大多数数学一样，光读公式是没啥大用处滴，你一定要尝试自己推导出这些结果（或者试一试课后练习也好啊）不然是没法深入理解这些玩意儿的)

21.6.1.1 那（传说中的）变分先验

高斯混合的似然函数就是还是熟悉的那个

$$p(z, X \mid \theta) = \prod_i \prod_k \pi_k^{z_{ik}} \mathcal{N}(x_i \mid \mu_k, \Lambda_k^{-1})^{z_{ik}} \quad (21.121)$$

此处有指示函数 $z_{ik} = 1$ 如若第 i 个样本属于类别 k, 反之则有 $z_{ik} = 0$。

我们假设有如下的因子化的共轭先验

$$p(\theta) = \mathrm{Dir}(\pi \mid \alpha_0)\pi_k \mathcal{N}(\mu_k \mid m_0, (\beta_0 \Lambda_k)^{-1})\mathrm{Wi}(\Lambda_k \mid L_0, \nu_0) \quad (21.122)$$

此处 Λ_k 是类别 k 的精度矩阵。下标 0 对应着先验参数，并假设所有的类别共享同样的先验参数。对于混合权重，则使用对称先验 $\boldsymbol{\alpha}_0 = \alpha_0 \mathbf{1}$。

该模型后验分布 $p(z, \theta \mid \mathcal{D})$ 的确切形式是由 K^N 个对应所有可能分类/标注方法 $\{z\}$ 的分布混合而成的。此处仅在其中某个标注模态附近做近似。考虑基本的变贝近似后验:

$$p(\theta, z_{1:N} \mid D) \approx q(\theta) \prod_i q(z_i) \quad (21.123)$$

此处我们还没有选定函数 q 的具体形式，它们会由似然和先验的形式共同决定。接下来我们会证明有如下的最优形式

$$q(z, \theta) = q(z \mid \theta) q(\theta) = \left[\prod_i \mathrm{Cat}(z_i \mid r_i) \right] \cdot \quad (21.124) \quad (45)$$

$$\left[\mathrm{Dir}(\pi \mid \alpha) \prod_k \mathcal{N}(\mu_k \mid m_k, (\beta_k \Lambda_k)^{-1}) \mathrm{Wi}(\Lambda_k \mid L_k, \nu_k) \right] \quad (21.125) \quad (46)$$

（注意上式中不含下标 0 因而都是后验而非先验的参数）。接下来会给出的这些变分参数的更新公式。

21.6.1.2 隐变量后验 $q(z)$ 的推导（变分期望步）

后验期望 $q(z)$ 的形式可以通过考虑数据似然函数：忽略那些不含 z 的那些项目，并将剩下来的项目对除去 z 的所有隐变量求期望，亦即

$$\log q(z) = \mathbb{E}_{q(\theta)}[\log p(x, z, \theta)] + \text{const} \quad (21.126) \tag{47}$$

$$= \sum_k \sum_i z_{ik} + \text{const} \quad (21.127) \tag{48}$$

并定义

$$\log \rho_{ik} \triangleq \mathbb{E}_{q(\theta)}[\log \pi_k] + \frac{1}{2}\mathbb{E}_{q(\theta)}[\log |\Lambda_k|] - \frac{D}{2}\log(2\pi) \quad - \frac{1}{2}\mathbb{E}_{q(\theta)}[(x_i - \mu_k)^T \Lambda_k (x_i - \mu_k)] \quad (21.128) \tag{49}$$

由于已经有 $q(\pi) = \text{Dir}(\pi)$，可以得出

$$\log \tilde{\pi} \triangleq \mathbb{E}[\log \pi_k] = \psi(\alpha_k) - \psi(\sum_{k'} \alpha_{k'}) \quad (21.129)$$

此处 $\psi()$ 是双伽马函数。（详细推导见 Exer21.5**??**）接下来利用如下事实

$$q(\mu_k, \Lambda_k) = \mathcal{N}(\mu_k \mid m_k, (\beta_k \Lambda_k)^{-1}) \text{Wi}(\Lambda_k \mid L_k, \nu_k) \quad (21.130)$$

然后得出

$$\log \tilde{\Lambda_k} \triangleq \mathbb{E}[\log |\Lambda_k|] = \sum_{j=1}^{D} \psi\left(\frac{v_k + 1 - j}{2}\right) + D\log 2 + \log |\Lambda_k| \quad (21.131)$$

最后，对二次项求期望可以得出

$$\mathbb{E}[(x_i - \mu_k)^T \Lambda_k (x_i - \mu_k)] = D\beta_k^{-1} + \nu_k (x_i - m_k)^T \Lambda_k (x_i - m_k) \quad (21.132)$$

综合考虑以上种种，得出

$$r_{ik} \propto \tilde{\pi_k} \tilde{\Lambda}_k^{\frac{1}{2}} \exp\left(-\frac{D}{2\beta_k} - \frac{\nu_k}{2}(x_i - m_k)^T \Lambda_k (x_i - m_k)\right) \quad (21.133)$$

将以上形式与常用的期最算法作对比

$$r_{ik}^{EM} \propto \hat{\pi}_k |\hat{\Lambda}|_k^{1/2} \exp\left(-\frac{1}{2}(x_i - \hat{\mu}_k)^T \hat{\Lambda}_k (x_i - \hat{\mu}_k)\right) \quad (21.134)$$

两者的区别会晚些在 Sec21.6.1.7**??**中探讨。

21.6.1.3 参变量后验分布 $q(\theta)$ 的推导（变分最大化步）

按照平均场的标准菜谱，你可以炒出

$$\log q(\theta) = \log p(\pi) + \sum_k \log p(\mu_k, \Lambda_k) + \sum_i \mathbb{E}_{q(z)}[\log p(z_i \mid \pi)] \quad + \sum_k \sum_i \mathbb{E}_{q(z)}[z_{ik}] \log \mathcal{N}(x_i \mid \mu_k, \Lambda_k^{-1}) \quad (50)$$

这玩意可以因子分解成

$$q(\theta) = q(\pi) \prod_k q(\mu_k, \Lambda_k) \quad (21.136)$$

收集所有含 π 的项可以得出

$$\log q(\pi) = (\alpha_0 - 1) \sum_k \log \pi_k + \sum_k \sum_i r_{ik} \log \pi_k + \text{const} \quad (21.137)$$

两边取指数，可以看出这是一个狄利克雷分布 (?-_-)

$$q(\pi) = \text{Dir}(\pi \mid \alpha) \quad (21.138) \quad (51)$$

$$\alpha_k = \alpha_0 + N_k \quad (21.139) \quad (52)$$

$$N_k = \sum_i r_{ik} \quad (21.138) \quad (53)$$

收集所有含 μ_k, Λ_k 的项目，可以得出

$$q(\mu_k, \Lambda_k) = \mathcal{N}(\mu_k \mid m_k, (\beta_k \Lambda_k)^{-1}) \mathrm{Wi}(\Lambda_k \mid L_k, \nu_k) \tag{21.141}$$

(54)

$$\beta_k = \beta_0 + N_k \tag{21.142}$$

(55)

$$m_k = (\beta_0 m_0 + N_k \bar{x_k})/\beta_k \tag{21.143}$$

(56)

$$L_k^{-1} = L_0^{-1} + N_k S_k + \frac{\beta_0 N_k}{\beta_0 + N_k} (\bar{x}_k - m_0)(\bar{x}_k - m_0)^T \tag{21.144}$$

(57)

$$\nu_k = \nu_0 + N_k + 1 \tag{21.145}$$

(58)

$$\bar{x}_k = \frac{1}{N_k} \sum_i r_{ik} x_i \tag{21.146}$$

(59)

$$S_k = \frac{1}{N_k} \sum_i r_{ik} (x_i - \bar{x}_k)(x_i - \bar{x}_k)^T \tag{21.147}$$

(60)

以上结果和最大似然后验点估计 (见 Sec11.4.2.8**??**) 的最大化步是很相似的，不过这里我们计算的是参变量 θ 之后验分布的超参数，而不是最大似然点估计。

21.6.1.4 边际似然的下界

这个算法设法最大化的是如下的似然函数下界

$$\mathcal{L} = \sum_z \int q(z, \theta) \log \frac{p(x, z, \theta)}{q(z, \theta)} d\theta \leq \log p(\mathcal{D}) \tag{21.148}$$

这个量应该随着每次迭代单调增加，如图 Fig21.7**??**所示。不幸的是，这个界限的推导过程有些杂乱，因为需要同时取未正规化的对数后验的期望并计算 q 的熵。该函数的推导细节 (其实和 Sec21.5.1.6**??**很相似) 留作习题 Exer21.4**??**。

21.6.1.5 后验预测分布

我们已经证得近似后验有如下形式

$$q(\theta) = \mathrm{Dir}(\pi \mid \alpha) \prod_k \mathcal{N}(\mu_k \mid m_k, (\beta_k \Lambda_k)^{-1}) \mathrm{Wi}(\Lambda_k \mid L_k, \nu_k) \quad (21.149)$$

因此后验预测分布可以参照 Sec4.6.3.6??的结果作如下近似

$$\begin{aligned} p(x \mid \mathcal{D}) &\approx \sum_z \int p(x \mid z, \theta) p(z \mid \theta) q(\theta) d\theta && (21.150) \quad (61) \\ &= \sum_k \int \pi_k \mathcal{N}(x \mid \mu_k, \Lambda_k^{-1}) q(\theta) d\theta && (21.151) \quad (62) \\ &= \sum_k \frac{\alpha_k}{\sum_{k'} \alpha_{k'}} \mathcal{T}(x \mid m_k, M_k, \nu_k + 1 - D) && (21.152) \quad (63) \\ M_k &= \frac{(v_k + 1 - D)\beta_k}{1 + \beta_k} L_k && (21.153) \quad (64) \end{aligned}$$

这实际上是一个学生 t-分布的加权混合。如果我们现在改作一个Plug-In 近似，则会得出高斯分布的加权和。

变贝期最的模型选择

为变贝模型选择分类数 K 的最简单的方法是多拟合几个模型，然后用对数边际似然的变分下界 $\mathcal{L}(K) \leq \log p(\mathcal{D} \mid K)$ 来近似 $p(K \mid \mathcal{D})$

$$p(K \mid \mathcal{D}) \approx \frac{\exp^{\mathcal{L}(K)}}{\sum_{K'} \exp^{\mathcal{L}(K')}} \quad (21.154)$$

实际上，这个下界还要考虑到参数的不可分辨性而做一些修改 (Sec11.3.1??)。特别地，尽管分贝能够近似参变量后验附近的概率密度 K，它事实上只能对付其中一个局域模态。对于 K 组分的混合分布，有 $K!$ 个由于标注的可置换性而等价的模态。因此我们应该使用如下修正: $\log p(\mathcal{D} \mid K) \approx \mathcal{L} + \log(K!)$。

21.6.1.7 变贝期最稀疏性的自动产生

尽管变贝提供了一个还凑活的关于边际似然的近似 (好过 BIC,???)，它仍然需要为多个分类数 K 各自拟合一个模型。另外一个选择是直接拟合仅仅一个 K 数巨大而 α_0 巨小的模型 $\alpha_0 \ll 1$。从图 2.14d\ref{fig:2.14d] 可以

看出，此时混合权重 π 的先验在单纯形的顶点附近有“尖峰”，因而会偏爱一个稀疏的混合权重。

在常规期最算法中，混合权重的最大后验点估计具有形式 $\hat{\pi}^k \propto (\alpha_k - 1)$，此处 $\alpha_k = \alpha_0 + N_k$。不幸的是，这玩意在 $\alpha_0 = 0$ 且 $N_k = 0$ 会取到负值 (**???**)。不过呢，在变贝期最中相应的后验估计是

$$\tilde{\pi}_k = \frac{\exp[\Psi(\alpha_k)]}{\exp[\Psi(\sum_{k'} \alpha_{k'})]} \quad (21.155)$$

对于 $x > 1$ 有近似 $\exp \Psi(x) \approx x - 0.5$。因而如果 $\alpha_k = 0$，那么就相当于我们在计算 $\tilde{\pi}_k$ 的时候从后验个数中减去了 0.5。这个效果对于没几个有分量成员的的小分类来说是更加严重的 (就像累退税那样)。其最终结果就是，随着不断地迭代，小分类变得越来越空而成员众多的分类变得越来越大。这也被称作是马太效应，这在 Sec25.2**??**讨论狄利克雷过程混合模型的时候会被再次提起。

这个自动剪枝方法在图 21.8**??**中进行了展示。一个 6 分类的高斯分布被用来拟合 OldFaithful 数据集，但是数据实际上只“需要”两个分类，因而多出来的那些分类就被“扼杀”了。在这个例子里，我们采用了 $\alpha_0 = 0.001$。如果我们用一个更大的 α_0，我们不一定会观察到稀疏化现象。在图 21.9**??**中我们展示了 $q(\alpha \mid \mathcal{D})$ 在不同迭代步的形状。可以见得多余的成分逐渐灭绝了。这相较于对分类数 K 的离散搜索是一个更有效率的替代方案。