28 变分自编码器（二）：从贝叶斯观点出发

Mar By 苏剑林 | 2018-03-28 | 76866位读者

源起

前几天写了博文《变分自编码器（一）：原来是这么一回事》，从一种比较通俗的观点来理解变分自编码器（VAE），在那篇文章的视角中，VAE跟普通的自编码器差别不大，无非是多加了噪声并对噪声做了约束。然而，当初我想要弄懂VAE的初衷，是想看看究竟贝叶斯学派的概率图模型究竟是如何与深度学习结合来发挥作用的，如果仅仅是得到一个通俗的理解，那显然是不够的。

所以我对VAE继续思考了几天，试图用更一般的、概率化的语言来把VAE说清楚。事实上，这种思考也能回答通俗理解中无法解答的问题，比如重构损失用MSE好还是交叉熵好、重构损失和KL损失应该怎么平衡，等等。

建议在阅读《变分自编码器（一）：原来是这么一回事》后对本文进行阅读，本文在内容上尽量不与前文重复。

准备

在进入对VAE的描述之前，我觉得有必要把一些概念性的内容讲一下。

数值计算vs采样计算

对于不是很熟悉概率统计的读者，容易混淆的两个概念应该是数值计算和采样计算，也有读者在《三味Capsule：矩阵Capsule与EM路由》出现过同样的疑惑。比如已知概率密度函数$p(x)$，那么x的期望也就定义为

$$\mathbb{E}[x] = \int xp(x)dx \tag{1}$$

如果要对它进行数值计算，也就是数值积分，那么可以选若干个有代表性的点$x_0 < x_1 < x_2 < \cdots < x_n$，然后得到

$$\mathbb{E}[x] \approx \sum_{i=1}^{n} x_i p(x_i)\left(x_i - x_{i-1}\right) \tag{2}$$

这里不讨论“有代表性”是什么意思，也不讨论提高数值计算精度的方法。这样写出来，是为了跟采样计算对比。如果从$p(x)$中采样若干个点$x_1, x_2, \ldots, x_n$，那么我们有

$$\mathbb{E}[x] \approx \frac{1}{n}\sum_{i=1}^{n} x_i, \quad x_i \sim p(x) \tag{3}$$

我们可以比较(2)跟(3)，它们的主要区别是(2)中包含了概率的计算而(3)中仅有x的计算，这是因为**在(3)中x_i是从$p(x)$中依概率采样出来的，概率大的x_i出现的次数也多，所以可以说采样的结果已经包含了$p(x)$在里边，就不用再乘以$p(x_i)$了。**

更一般地，我们可以写出

$$\mathbb{E}_{x\sim p(x)}[f(x)] = \int f(x)p(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(x_i), \quad x_i \sim p(x) \tag{4}$$

这就是蒙特卡洛模拟的基础。

KL散度及变分

我们通常用KL散度来度量两个概率分布$p(x)$和$q(x)$之间的差异，定义为

$$KL\Big(p(x)\Big\|q(x)\Big) = \int p(x)\ln\frac{p(x)}{q(x)}dx = \mathbb{E}_{x\sim p(x)}\left[\ln\frac{p(x)}{q(x)}\right] \tag{5}$$

KL散度的主要性质是非负性，如果固定$p(x)$，那么$KL\Big(p(x)\Big\|q(x)\Big) = 0 \Leftrightarrow p(x) = q(x)$；如果固定$q(x)$，同样有$KL\Big(p(x)\Big\|q(x)\Big) = 0 \Leftrightarrow p(x) = q(x)$，也就是不管固定哪一个，最小化KL散度的结果都是两者尽可能相等。这一点的严格证明要用到变分法，而事实上VAE中的V（变分）就是因为VAE的推导就是因为用到了KL散度（进而也包含了变分法）。

当然，KL散度有一个比较明显的问题，就是当$q(x)$在某个区域等于0，而$p(x)$在该区域不等于0，那么KL散度就出现无穷大。这是KL散度的固有问题，我们只能想办法规避它，比如隐变量的先验分布我们用高斯分布而不是均匀分布，原因便在此，这一点我们在前文《变分自编码器（一）：原来是这么一回事》中也提到过了。

顺便说点题外话，度量两个概率分布之间的差异只有KL散度吗？当然不是，我们可以看维基百科的Statistical Distance一节，里边介绍了不少分布距离，比如有一个很漂亮的度量，我们称之为巴氏距离（Bhattacharyya distance），定义为

$$D_B\Big(p(x),q(x)\Big) = -\ln\int\sqrt{p(x)q(x)}dx \tag{6}$$

这个距离不仅对称，还没有KL散度的无穷大问题。然而我们还是选用KL散度，因为我们不仅要理论上的漂亮，还要实践上的可行，KL散度可以写成期望的形式，这允许我们对其进行采样计算，相反，巴氏距离就没那么容易了，读者要是想把下面计算过程中的KL散度替换成巴氏距离，就会发现寸步难行了。

本文的符号表

讲解VAE免不了出现大量的公式和符号，这里将部分式子的含义提前列举如下：

x_k, z_k	表示随机变量x, z的第k个样本
$x_{(k)}, z_{(k)}$	表示多元变量x, z的第k个分量
$\mathbb{E}_{x\sim p(x)}[f(x)]$	表示对$f(x)$算期望，其中x的分布为$p(x)$
$KL\Big(p(x)\Big\|q(x)\Big)$	两个分布的KL散度
$\|x\|^2$	向量x的l^2范数，也就是我们通常说的模长的平方
$\mathcal{L}$	本文的损失函数的符号
D, d	D是输入x的维度，d是隐变量z的维度

框架

这里通过直接对联合分布进行近似的方式，简明快捷地给出了VAE的理论框架。

直面联合分布

出发点依然没变，这里再重述一下。首先我们有一批数据样本$\{x_1,\dots,x_n\}$，其整体用x来描述，我们希望借助隐变量z描述x的分布$\tilde{p}(x)$：

$$q(x) = \int q(x|z)q(z)dz, \quad q(x,z) = q(x|z)q(z) \tag{7}$$

这里$q(z)$是先验分布（标准正态分布），目的是希望$q(x)$能逼近$\tilde{p}(x)$。这样（理论上）我们既描述了$\tilde{p}(x)$，又得到了生成模型$q(x|z)$，一举两得。

接下来就是利用KL散度进行近似。**但我一直搞不明白的是，为什么从原作《Auto-Encoding Variational Bayes》开始，VAE的教程就聚焦于后验分布$p(z|x)$的描述？也许是受了EM算法的影响，这个问题上不能应用EM算法，就是因为后验分布$p(z|x)$难以计算，所以VAE的作者就聚焦于$p(z|x)$的推导。**

但事实上，直接来对$p(x,z)$进行近似是最为干脆的。具体来说，定义$p(x,z)=\tilde{p}(x)p(z|x)$，我们设想用一个联合概率分布$q(x,z)$来逼近$p(x,z)$，那么我们用KL散度来看它们的距离：

$$KL\Big(p(x,z)\Big\|q(x,z)\Big)=\iint p(x,z)\ln\frac{p(x,z)}{q(x,z)}dzdx \tag{8}$$

KL散度是我们的终极目标，因为我们希望两个分布越接近越好，所以KL散度越小越好。

于是我们有

$$\begin{aligned}KL\Big(p(x,z)\Big\|q(x,z)\Big)=&\int\tilde{p}(x)\left[\int p(z|x)\ln\frac{\tilde{p}(x)p(z|x)}{q(x,z)}dz\right]dx\\ =&\mathbb{E}_{x\sim\tilde{p}(x)}\left[\int p(z|x)\ln\frac{\tilde{p}(x)p(z|x)}{q(x,z)}dz\right]\end{aligned} \tag{9}$$

这样一来利用(4)式，把各个x_i代入就可以进行计算了，这个式子还可以进一步简化，因为$\ln\frac{\tilde{p}(x)p(z|x)}{q(x,z)}=\ln\tilde{p}(x)+\ln\frac{p(z|x)}{q(x,z)}$，而

$$\begin{aligned}\mathbb{E}_{x\sim\tilde{p}(x)}\left[\int p(z|x)\ln\tilde{p}(x)dz\right]=&\mathbb{E}_{x\sim\tilde{p}(x)}\left[\ln\tilde{p}(x)\int p(z|x)dz\right]\\ =&\mathbb{E}_{x\sim\tilde{p}(x)}\Big[\ln\tilde{p}(x)\Big]\end{aligned} \tag{10}$$

注意这里的$\tilde{p}(x)$是根据样本$x_1,x_2,\ldots,x_n$确定的关于x的先验分布，尽管我们不一定能准确写出它的形式，但它是确定的、存在的，因此这一项只是一个常数，所以可以写出

$$\mathcal{L}=KL\Big(p(x,z)\Big\|q(x,z)\Big)-\text{常数}=\mathbb{E}_{x\sim\tilde{p}(x)}\left[\int p(z|x)\ln\frac{p(z|x)}{q(x,z)}dz\right] \tag{11}$$

目前最小化$KL\Big(p(x,z)\Big\|q(x,z)\Big)$也就等价于最小化$\mathcal{L}$。注意减去的常数为$\mathbb{E}_{x\sim\tilde{p}(x)}\Big[\ln\tilde{p}(x)\Big]$，所以$\mathcal{L}$拥有下界$-\mathbb{E}_{x\sim\tilde{p}(x)}\Big[\ln\tilde{p}(x)\Big]$～注意到$\tilde{p}(x)$不一定是概率，在连续情形时$\tilde{p}(x)$是概率密度，它可以大于1也可以小于1，所以$-\mathbb{E}_{x\sim\tilde{p}(x)}\Big[\ln\tilde{p}(x)\Big]$不一定是非负，即loss可能是负数。

你的VAE已经送达

到这里，我们回顾初衷——为了得到生成模型，所以我们把$q(x,z)$写成$q(x|z)q(z)$，于是就有

$$\begin{aligned}\mathcal{L}=&\mathbb{E}_{x\sim\tilde{p}(x)}\left[\int p(z|x)\ln\frac{p(z|x)}{q(x|z)q(z)}dz\right]\\ =&\mathbb{E}_{x\sim\tilde{p}(x)}\left[-\int p(z|x)\ln q(x|z)dz+\int p(z|x)\ln\frac{p(z|x)}{q(z)}dz\right]\end{aligned} \tag{12}$$

再简明一点，那就是

$$\begin{aligned}\mathcal{L} =&\mathbb{E}_{x\sim\tilde{p}(x)}\left[\mathbb{E}_{z\sim p(z|x)}\left[-\ln q(x|z)\right]+\mathbb{E}_{z\sim p(z|x)}\left[\ln\frac{p(z|x)}{q(z)}\right]\right]\\ =&\mathbb{E}_{x\sim\tilde{p}(x)}\left[\mathbb{E}_{z\sim p(z|x)}\left[-\ln q(x|z)\right]+KL\Big(p(z|x)\Big\|q(z)\Big)\right]\end{aligned}\tag{13}$$

看，括号内的不就是VAE的损失函数嘛？只不过我们换了个符号而已。我们就是要想办法找到适当的$q(x|z)$和$q(z)$使得$\mathcal{L}$最小化。

再回顾一下整个过程，我们几乎都没做什么“让人难以想到”的形式变换，但VAE就出来了。所以，没有必要去对后验分布进行分析，直面联合分布，我们能更快捷地到达终点。

不能搞分裂～

鉴于(13)式的特点，我们也许会将$\mathcal{L}$分开为两部分看：$\mathbb{E}_{z\sim p(z|x)}\left[-\ln q(x|z)\right]$的期望和$KL\Big(p(z|x)\Big\|q(z)\Big)$的期望，并且认为问题变成了两个loss的分别最小化。

然而这种看法是不妥的，因为$KL\Big(p(z|x)\Big\|q(z)\Big)=0$意味着$z$没有任何辨识度，所以$-\ln q(x|z)$不可能小（预测不准），而如果$-\ln q(x|z)$小则$q(x|z)$大，预测准确，这时候$p(z|x)$不会太随机，即$KL\Big(p(z|x)\Big\|q(z)\Big)$不会小，所以这两部分的loss其实是相互拮抗的。所以，$\mathcal{L}$不能割裂来看，而是要整体来看，整个的$\mathcal{L}$越小模型就越接近收敛，而不能只单独观察某一部分的loss。

事实上，这正是GAN模型中梦寐以求的——有一个总指标能够指示生成模型的训练进程，在VAE模型中天然就具备了这种能力了，而GAN中要到WGAN才有这么一个指标～

实验

截止上面的内容，其实我们已经完成了VAE整体的理论构建。但为了要将它付诸于实验，还需要做一些工作。事实上原论文《Auto-Encoding Variational Bayes》也在这部分做了比较充分的展开，但遗憾的是，网上很多VAE教程都只是推导到(13)式就没有细说了。

后验分布近似

现在$q(z),q(x|z),p(z|x)$全都是未知的，连形式都还没确定，而为了实验，就得把(13)式的每一项都明确写出来。

首先，为了便于采样，我们假设$z\sim N(0,I)$，即标准的多元正态分布，这就解决了$q(z)$。那$q(x|z),p(z|x)$呢？**一股脑用神经网络拟合吧。**

注：本来如果已知$q(x|z)$和$q(z)$，那么$p(z|x)$最合理的估计应该是：

$$\hat{p}(z|x)=q(z|x)=\frac{q(x|z)q(z)}{q(x)}=\frac{q(x|z)q(z)}{\int q(x|z)q(z)dz}\tag{14}$$

这其实就是EM算法中的后验概率估计的步骤，具体可以参考《从最大似然到EM算法：一致的理解方式》。但事实上，分母的积分几乎不可能完成，因此这是行不通的。所以干脆用一般的网络去近似它，这样不一定能达到最优，但终究是一个可用的近似。

具体来说，我们假设$p(z|x)$也是（各分量独立的）正态分布，其均值和方差由x来决定，这个“决定”，就是一个神经网络：

$$p(z|x) = \frac{1}{\prod_{k=1}^{d}\sqrt{2\pi\sigma_{(k)}^2(x)}}\exp\left(-\frac{1}{2}\left\|\frac{z-\mu(x)}{\sigma(x)}\right\|^2\right) \tag{15}$$

这里的$\mu(x), \sigma^2(x)$是输入为x、输出分别为均值和方差的神经网络，其中$\mu(x)$就起到了类似encoder的作用。既然假定了高斯分布，那么(13)式中的KL散度这一项就可以先算出来：

$$KL\Big(p(z|x)\Big\|q(z)\Big) = \frac{1}{2}\sum_{k=1}^{d}\Big(\mu_{(k)}^2(x) + \sigma_{(k)}^2(x) - \ln\sigma_{(k)}^2(x) - 1\Big) \tag{16}$$

也就是我们所说的KL loss，这在上一篇文章已经给出。

生成模型近似

现在只剩生成模型部分$q(x|z)$了，该选什么分布呢？论文《Auto-Encoding Variational Bayes》给出了两种候选方案：伯努利分布或正态分布。

什么？又是正态分布？是不是太过简化了？然而并没有办法，因为我们要构造一个分布，而不是任意一个函数，既然是分布就得满足归一化的要求，而要满足归一化，又要容易算，我们还真没多少选择。

伯努利分布模型

首先来看伯努利分布，众所周知它其实就是一个二元分布：

$$p(\xi) = \begin{cases}\rho, & \xi = 1;\\ 1-\rho, & \xi = 0\end{cases} \tag{17}$$

所以伯努利分布只适用于x是一个多元的二值向量的情况，比如x是二值图像时（mnist可以看成是这种情况）。这种情况下，我们用神经网络$\rho(z)$来算参数ρ，从而得到

$$q(x|z) = \prod_{k=1}^{D}\Big(\rho_{(k)}(z)\Big)^{x(k)}\Big(1-\rho_{(k)}(z)\Big)^{1-x(k)} \tag{18}$$

这时候可以算出

$$-\ln q(x|z) = \sum_{k=1}^{D}\Big[-x_{(k)}\ln\rho_{(k)}(z) - (1-x_{(k)})\ln\Big(1-\rho_{(k)}(z)\Big)\Big] \tag{19}$$

这表明$\rho(z)$要压缩到0～1之间（比如用sigmoid激活），然后用交叉熵作为损失函数，这里$\rho(z)$就起到了类似decoder的作用。

正态分布模型

然后是正态分布，这跟$p(z|x)$是一样的，只不过x, z交换了位置：

$$q(x|z) = \frac{1}{\prod_{k=1}^{D}\sqrt{2\pi\tilde{\sigma}_{(k)}^2(z)}}\exp\left(-\frac{1}{2}\left\|\frac{x-\tilde{\mu}(z)}{\tilde{\sigma}(z)}\right\|^2\right) \tag{20}$$

这里的$\tilde{\mu}(z), \tilde{\sigma}^2(z)$是输入为$z$、输出分别为均值和方差的神经网络，$\tilde{\mu}(z)$就起到了decoder的作用。于是

$$-\ln q(x|z) = \frac{1}{2}\left\|\frac{x-\tilde{\mu}(z)}{\tilde{\sigma}(z)}\right\|^2 + \frac{D}{2}\ln 2\pi + \frac{1}{2}\sum_{k=1}^{D}\ln\tilde{\sigma}_{(k)}^2(z) \tag{21}$$

很多时候我们会固定方差为一个常数$\tilde{\sigma}^2$，这时候

$$-\ln q(x|z) \sim \frac{1}{2\tilde{\sigma}^2}\left\|x-\tilde{\mu}(z)\right\|^2 \tag{22}$$

这就出现了MSE损失函数。

所以现在就清楚了，对于二值数据，我们可以对decoder用sigmoid函数激活，然后用交叉熵作为损失函数，这对应于$q(x|z)$为伯努利分布；而对于一般数据，我们用MSE作为损失函数，这对应于$q(x|z)$为固定方差的正态分布。

采样计算技巧

前一节做了那么多的事情，无非是希望能(13)式明确地写下来。当我们假设$p(z|x)$和$q(z)$都是正态分布时，(13)式的KL散度部分就已经算出来了，结果是(16)式；当我们假设$q(x|z)$是伯努利分布或者高斯分布时，$-\ln q(x|z)$也能算出来了。现在缺什么呢？

采样！

$p(z|x)$的作用分两部分，一部分是用来算$KL\Big(p(z|x)\Big\|q(z)\Big)$，另一部分是用来算$\mathbb{E}_{z\sim p(z|x)}\big[-\ln q(x|z)\big]$的，而$\mathbb{E}_{z\sim p(z|x)}\big[-\ln q(x|z)\big]$就意味着

$$-\frac{1}{n}\sum_{i=1}^{n}\ln q(x|z_i), \quad z_i \sim p(z|x) \tag{23}$$

我们已经假定了$p(z|x)$是正态分布，均值和方差由模型来算，这样一来，借助“重参数技巧”就可以完成采样。

但是采样多少个才适合呢？VAE非常直接了当：一个！所以这时候(13)式就变得非常简单了：

$$\mathcal{L} = \mathbb{E}_{x\sim\tilde{p}(x)}\left[-\ln q(x|z) + KL\Big(p(z|x)\Big\|q(z)\Big)\right], \quad z\sim p(z|x) \tag{24}$$

该式中的每一项，可以在把(16), (19), (21), (22)式找到。注意对于一个batch中的每个x，都需要从$p(z|x)$采样一个“专属”于x的z出来才去算$-\ln q(x|z)$。而正因为VAE在$p(z|x)$这里只采样了一个样本，所以它看起来就跟普通的AE差不多了。

那么最后的问题就是采样一个究竟够了吗？事实上我们会运行多个epoch，每次的隐变量都是随机生成的，因此当epoch数足够多时，事实上是可以保证采样的充分性的。我也实验过采样多个的情形，感觉生成的样本并没有明显变化。

致敬

这篇文章从贝叶斯理论的角度出发，对VAE的整体流程做了一个梳理。用这种角度考察的时候，我们心里需要紧抓住两个点：“分布”和“采样”——写出分布形式，并且通过采样来简化过程。

简单来说，由于直接描述复杂分布是难以做到的，所以我们通过引入隐变量来将它变成条件分布的叠加。而这时候我们对隐变量的分布和条件分布都可以做适当的简化（比如都假设为正态分布），并且在条件分布的参数可以跟深度学习模型结合起来（用深度学习来算隐变量的参数），至此，“深度概率图模型”就可见一斑了。

让我们一起致敬贝叶斯大神，以及众多研究概率图模型的大牛，他们都是真正的勇者。