17 May

变分自编码器（七）：球面上的VAE（vMF-VAE）

By 苏剑林 | 2021-05-17 | 134433位读者 引用

在《变分自编码器（五）：VAE + BN = 更好的VAE》中，我们讲到了NLP中训练VAE时常见的KL散度消失现象，并且提到了通过BN来使得KL散度项有一个正的下界，从而保证KL散度项不会消失。事实上，早在2018年的时候，就有类似思想的工作就被提出了，它们是通过在VAE中改用新的先验分布和后验分布，来使得KL散度项有一个正的下界。

该思路出现在2018年的两篇相近的论文中，分别是《Hyperspherical Variational Auto-Encoders》和《Spherical Latent Spaces for Stable Variational Autoencoders》，它们都是用定义在超球面的von Mises–Fisher（vMF）分布来构建先后验分布。某种程度上来说，该分布比我们常用的高斯分布还更简单和有趣～

KL散度消失

我们知道，VAE的训练目标是

$$\mathcal{L} = \mathbb{E}_{x\sim \tilde{p}(x)}\left[\mathbb{E}_{z\sim p(z|x)}\big[-\log q(x|z)\big]+KL\Big(p(z|x)\Big\Vert q(z)\Big)\right]\tag{1}$$

其中第一项是重构项，第二项是KL散度项，在《变分自编码器（一）：原来是这么一回事》中我们就说过，这两项某种意义上是“对抗”的，KL散度项的存在，会加大解码器利用编码信息的难度，如果KL散度项为0，那么说明解码器完全没有利用到编码器的信息。

在NLP中，输入和重构的对象是句子，为了保证效果，解码器一般用自回归模型。然而，自回归模型是非常强大的模型，强大到哪怕没有输入，也能完成训练（退化为无条件语言模型），而刚才我们说了，KL散度项会加大解码器利用编码信息的难度，所以解码器干脆弃之不用，这就出现了KL散度消失现象。

早期比较常见的应对方案是逐渐增加KL项的权重，以引导解码器去利用编码信息。现在比较流行的方案就是通过某些改动，直接让KL散度项有一个正的下界。将先后验分布换为vMF分布，就是这种方案的经典例子之一。

vMF分布

vMF分布是定义在$d-1$维超球面的分布，其样本空间为$S^{d-1}=\{x|x\in\mathbb{R}^d, \|x\|=1\}$，概率密度函数则为

$$p(x)=\frac{e^{\langle\xi,x\rangle}}{Z_{d,\|\xi\|}},\quad Z_{d,\|\xi\|}=\int_{S^{d-1}}e^{\langle\xi,x\rangle}dS^{d-1}\tag{2}$$

其中$\xi\in\mathbb{R}^d$是预先给定的参数向量。不难想象，这是S^{d-1}上一个以ξ为中心的分布，归一化因子写成$Z_{d,\|\xi\|}$的形式，意味着它只依赖于ξ的模长，这是由于各向同性导致的。由于这个特性，vMF分布更常见的记法是设$\mu=\xi/\|\xi\|,\kappa=\|\xi\|,C_{d,\kappa}=1/Z_{d,\|\xi\|}$，从而

$$p(x)=C_{d,\kappa}e^{\kappa\langle\mu,x\rangle}\tag{3}$$

这时候$\langle\mu,x\rangle$就是μ,x的夹角余弦，所以说，vMF分布实际上就是以余弦相似度为度量的一种分布。由于我们经常用余弦值来度量两个向量的相似度，因此基于vMF分布做出来的模型，通常更能满足我们的这个需求。当$\kappa=0$的时候，vMF分布是球面上的均匀分布。

从归一化因子$Z_{d,\|\xi\|}$的积分形式来看，它实际上也是vMF的母函数，从而vMF的各阶矩也可以通过$Z_{d,\|\xi\|}$来表达，比如一阶矩为

$$\mathbb{E}_{x\sim p(x)}[x]=\nabla_{\xi}\log Z_{d,\|\xi\|}=\frac{d\log Z_{d,\|\xi\|}}{d\|\xi\|}\frac{\xi}{\|\xi\|}\tag{4}$$

可以看到$\mathbb{E}_{x\sim p(x)}[x]$在方向上跟$\xi$一致。$Z_{d,\|\xi\|}$的精确形式可以算出来，但比较复杂，而且很多时候我们也不需要精确知道这个归一化因子，所以这里我们就不算了。

至于参数κ的含义，或许设$\tau = 1/\kappa$我们更好理解，此时$p(x) \sim e^{\langle \mu, x\rangle/\tau}$，熟悉能量模型的同学都知道，这里的$\tau$就是温度参数，如果$\tau$越小（$\kappa$越大），那么分布就越集中在$\mu$附近，反之则越分散（越接近球面上的均匀分布）。因此，κ也被形象地称为"凝聚度（concentration）"参数。

从vMF采样

对于vMF分布来说，需要解决的第一个难题是如何实现从它里边采样出具体的样本来。尤其是如果我们要将它应用到VAE中，那么这一步是至关重要的。

均匀分布

最简单是$\kappa = 0$的情形，也就是$d-1$维球面上的均匀分布，因为标准正态分布本来就是各向同性的，其概率密度正比于$e^{-\Vert x\Vert^2/2}$只依赖于模长，所以我们只需要从d为标准正态分布中采样一个z，然后让$x = z/\Vert z\Vert$就得到了球面上的均匀采样结果。

特殊方向

接着，对于$\kappa > 0$的情形，我们记$x = [x_1, x_2, \cdots, x_d]$，首先考虑一种特殊的情况：$\mu = [1, 0, \cdots, 0]$。事实上，由于各向同性的原因，很多时候我们都只需要考虑这个特殊情况，然后就可以平行地推广到一般情形。

此时概率密度正比于$e^{\kappa x_1}$，然后我们转换到球坐标系：

$$\left\{\begin{aligned}
x_1 &= \cos\varphi_1 \\
x_2 &= \sin\varphi_1\cos\varphi_2 \\
x_3 &= \sin\varphi_1\sin\varphi_2\cos\varphi_3 \\
&\vdots \\
x_{d-1} &= \sin\varphi_1\cdots\sin\varphi_{d-2}\cos\varphi_{d-1} \\
x_d &= \sin\varphi_1\cdots\sin\varphi_{d-2}\sin\varphi_{d-1}
\end{aligned}\right.\tag{5}$$

那么（超球坐标的积分变换，请直接参考"维基百科"）

$$\begin{aligned}e^{\kappa x_1}dS^{d-1} =&e^{\kappa\cos\varphi_1}\sin^{d-2}\varphi_1\sin^{d-3}\varphi_2\cdots\sin\varphi_{d-2}d\varphi_1 d\varphi_2\cdots d\varphi_{d-1}\\ =&\left(e^{\kappa\cos\varphi_1}\sin^{d-2}\varphi_1 d\varphi_1\right)\left(\sin^{d-3}\varphi_2\cdots\sin\varphi_{d-2}d\varphi_2\cdots d\varphi_{d-1}\right)\\ =&\left(e^{\kappa\cos\varphi_1}\sin^{d-2}\varphi_1 d\varphi_1\right)dS^{d-2}\end{aligned}\tag{6}$$

这个分解表明，从该vMF分布中采样，等价于先从概率密度正比于$e^{\kappa\cos\varphi_1}\sin^{d-2}\varphi_1$的分布采样一个$\varphi_1$，然后从$d-2$维超球面上均匀采样一个$d-1$维向量$\varepsilon=[\varepsilon_2,\varepsilon_3,\cdots,\varepsilon_d]$，通过如下方式组合成最终采样结果

$$x = [\cos\varphi_1, \varepsilon_2\sin\varphi_1, \varepsilon_3\sin\varphi_1,\cdots,\varepsilon_d\sin\varphi_1]\tag{7}$$

设$w=\cos\phi_1\in[-1,1]$，那么

$$\left|e^{\kappa\cos\varphi_1}\sin^{d-2}\varphi_1 d\varphi_1\right| = \left|e^{\kappa w}(1-w^2)^{(d-3)/2}dw\right|\tag{8}$$

所以我们主要研究从概率密度正比于$e^{\kappa w}(1-w^2)^{(d-3)/2}$的分布中采样。

然而，笔者所不理解的是，大多数涉及到vMF分布的论文，都采用了1994年的论文《Simulation of the von mises fisher distribution》提出的基于beta分布的拒绝采样方案，整个采样流程还是颇为复杂的。但现在都2021年了，对于一维分布的采样，居然还需要拒绝采样这么低效的方案？

事实上，对于任意一维分布$p(w)$，设它的累积概率函数为$\Phi(w)$，那么$w=\Phi^{-1}(\varepsilon),\varepsilon\sim U[0,1]$就是一个最方便通用的采样方案。可能有读者抗议说“累积概率函数不好算呀”、“它的逆函数更不好算呀”，但是在用代码实现采样的时候，我们压根就不需要知道$\Phi(w)$长啥样，只要直接数值计算就行了，参考实现如下：

```
import numpy as np

def sample_from_pw(size, kappa, dims, epsilon=1e-7):
    x = np.arange(-1 + epsilon, 1, epsilon)
    y = kappa * x + np.log(1 - x**2) * (dims - 3) / 2
    y = np.cumsum(np.exp(y - y.max()))
    y = y / y[-1]
    return np.interp(np.random.random(size), y, x)
```

这里的实现中，计算量最大的是变量y的计算，而一旦计算好之后，可以缓存下来，之后只需要执行最后一步来完成采样，其速度是非常快的。这样再怎么看，也比从beta分布中拒绝采样要简单方便吧。顺便说，实现上这里还用到了一个技巧，即先计算对数值，然后减去最大值，最后才算指数，这样可以防止溢出，哪怕κ成千上万，也可以成功计算。

一般情形

现在我们已经实现了从$\mu=[1,0,\cdots,0]$的vMF分布中采样了，我们可以将采样结果分解为

$$x = w\times\underbrace{[1,0,\cdots,0]}_{\text{参数向量}\mu} + \sqrt{1-w^2}\times\underbrace{[0,\varepsilon_2,\cdots,\varepsilon_d]}_{\substack{\text{与}\mu\text{正交的}d-2\text{维}\\ \text{超球面均匀采样}}} \tag{9}$$

同样由于各向同性的原因，对于一般的μ，采样结果依然具有同样的形式：

$$\begin{aligned}&x = w\mu + \sqrt{1-w^2}\nu\\ &w\sim e^{\kappa w}(1-w^2)^{(d-3)/2}\\ &\nu\sim\text{与}\mu\text{正交的}d-2\text{维超球面均匀分布}\end{aligned} \tag{10}$$

对于ν的采样，关键之处是与μ正交，这也不难实现，先从标准正态分布中采样一个d维向量z，然后保留与μ正交的分量并归一化即可：

$$\nu = \frac{\varepsilon - \langle\varepsilon,\mu\rangle\mu}{\Vert\varepsilon - \langle\varepsilon,\mu\rangle\mu\Vert},\quad \varepsilon\sim\mathcal{N}(0,1_d) \tag{11}$$

vMF-VAE

至此，我们可谓是已经完成了本篇文章最艰难的部分，剩下的构建vMF-VAE可谓是水到渠成了。vMF-VAE选用球面上的均匀分布（$\kappa=0$）作为先验分布$q(z)$，并将后验分布选取为vMF分布：

$$p(z|x) = C_{d,\kappa}e^{\kappa\langle\mu(x),z\rangle} \tag{12}$$

简单起见，我们将κ设为超参数（也可以理解为通过人工而不是梯度下降来更新这个参数），这样一来，$p(z|x)$的唯一参数来源就是$\mu(x)$了。此时我们可以计算KL散度项

$$\begin{aligned}\int p(z|x)\log\frac{p(z|x)}{q(z)}dz =& \int C_{d,\kappa}e^{\kappa\langle\mu(x),z\rangle}\left(\kappa\langle\mu(x),z\rangle + \log C_{d,\kappa} - \log C_{d,0}\right)dz\\ =& \kappa\left\langle\mu(x),\mathbb{E}_{z\sim p(z|x)}[z]\right\rangle + \log C_{d,\kappa} - \log C_{d,0}\end{aligned}$$

前面我们已经讨论过，vMF分布的均值方向跟$\mu(x)$一致，模长则只依赖于d和κ，所以代入上式后我们可以知道KL散度项只依赖于d和κ，当这两个参数被选定之后，那么它就是一个常数（根据KL散度的性质，当$\kappa\neq 0$时，它必然大于0），绝对不会出现KL散度消失现象了。

那么现在就剩下重构项了，我们需要用“重参数（Reparameterization）”来完成采样并保留梯度，在前面我们已经研究了vMF的采样过程，所以也不难实现，综合的流程为：

$$\begin{aligned}&\mathcal{L} = \Vert x - g(z)\Vert^2\\ &z = w\mu(x) + \sqrt{1-w^2}\nu\\ &w\sim e^{\kappa w}(1-w^2)^{(d-3)/2}\\ &\nu = \frac{\varepsilon - \langle\varepsilon,\mu\rangle\mu}{\Vert\varepsilon - \langle\varepsilon,\mu\rangle\mu\Vert}\\ &\varepsilon\sim\mathcal{N}(0,1_d)\end{aligned} \tag{14}$$

这里的重构loss以MSE为例，如果是句子重构，那么换用交叉熵就好。其中$\mu(x)$就是编码器，而$g(z)$就是解码器，由于KL散度项为常数，对优化没影响，所以vMF-VAE相比于普通的自编码器，只是多了一项稍微有点复杂的重参数操作（以及人工调整κ）而已，相比基于高斯分布的标准VAE可谓简化了不少了。

此外，从该流程我们也可以看出，除了“简单起见”之外，不将κ设为可训练还有一个主要原因，那就是κ关系到w的采样，而在w的采样过程中要保留κ的梯度是比较困难的。

参考实现

vMF-VAE的实现难度主要是重参数部分，也就还是从vMF分布中采样，而关键之处就是w的采样。前面我们已经给出了w的采样的numpy实现，但是在tf中未见类似`np.interp`的函数，因此不容易转换为纯tf的实现。当然，如果是torch或者tf2这种动态图框架，直接跟numpy的代码混合使用也无妨，但这里还是想构造一种比较通用的方案。

其实也不难，由于w只是一个一维变量，每步训练只需要用到`batch_size`个采样结果，所以我们完全可以事先用numpy函数采样好足够多（几十万）个w存好，然后训练的时候直接从这批采样好的结果随机抽就行了，参考实现如下：

```
def sampling(mu):
    """vMF分布重参数操作
    """
    dims = K.int_shape(mu)[-1]
    # 预先计算一批w
    epsilon = 1e-7
    x = np.arange(-1 + epsilon, 1, epsilon)
    y = kappa * x + np.log(1 - x**2) * (dims - 3) / 2
    y = np.cumsum(np.exp(y - y.max()))
    y = y / y[-1]
    W = K.constant(np.interp(np.random.random(10**6), y, x))
    # 实时采样w
    idxs = K.random_uniform(K.shape(mu[:, :1]), 0, 10**6, dty
    w = K.gather(W, idxs)
    # 实时采样z
    eps = K.random_normal(K.shape(mu))
    nu = eps - K.sum(eps * mu, axis=1, keepdims=True) * mu
    nu = K.l2_normalize(nu, axis=-1)
    return w * mu + (1 - w**2)**0.5 * nu
```

一个基于MNIST的完整例子可见：

https://github.com/bojone/vae/blob/master/vae_vmf_keras.py

至于vMF-VAE用于NLP的例子，我们日后有机会再分享。本文主要还是以理论介绍和简单演示为主～

文章小结

本文介绍了基于vMF分布的VAE实现，其主要难度在于vMF分布的采样。总的来说，vMF分布建立在余弦相似度度量之上，在某些方面的性质更符合我们的直观认知，将其用于VAE中，能够使得KL散度项为一个常数，从而防止了KL散度消失现象，并且简化了VAE结构。

如果您需要引用本文，请参考：

苏剑林. (May. 17, 2021). 《变分自编码器（七）：球面上的VAE（vMF-VAE）》[Blog post]. Retrieved from https://spaces.ac.cn/archives/8404

```
@online{kexuefm-8404,
        title={变分自编码器（七）：球面上的VAE（vMF-VAE）},
        author={苏剑林},
        year={2021},
        month={May},
        url={\url{https://spaces.ac.cn/archives/8404}},
}
```