生成模型读书笔记十

2021年1月13日　14:58

生成模型对比和总结

总结及分析：GAN，VAE以及其他生成模型之间的关系

从变分推断的角度来统一本文所介绍的模型，这一段有点难理解，把参考资料贴出来，日后可能有新的理解：https://kexue.fm/archives/5716/comment-page-1#comments，On Unifying Deep Generative Models，https://www.zhihu.com/question/40797593

a. 回顾变分推断

当数据分布包含观测变量x和隐变量z时，推断就是指拟合x和z的联合分布p(x,z)或者z的后验分布p(z|x)，希望能够从观测数据推测出能够解释这个数据的隐变量。在推断中，因为真实分布p比较难解，所以通常会用一个假设的分布q来近似它，这个分布q就是变分分布。变分推断的主要过程就是最小化q和p的KL距离。在"概率模型的推断一节"中我们举了个性别和身高的例子，简而言之就是，推断：观测数据 ⟹ 隐变量；生成：隐变量 ⟹ 数据

b. 回顾EM

EM常用于我们明确知道分布族，知道对应的概率密度函数，但是不知道当中具体的参数，需要在有隐变量的情况下算出其中参数的情形。通常会设$p_\theta(x,y)$为带有未知参数的属于某个已知分布族的分布函数。

回顾一下EM一节中从KL散度推导的内容，我们将$p_\theta(x,y)$写成q(x, z)，那么就可以与变分推断定义的符号一致，q(x, z)就是变分函数，我们用q来近似真实的分布p(x, z)，下面将推导过程复习一遍

$$
\begin{aligned}
KL(p(x,z)\,\|\,q(x,z)) &= \iint p(x,z)log\frac{p(x,z)}{q(x,z)}dxdz \\
&= \iint p(z|x)p(x)log\frac{p(z|x)*p(x)}{q(x|z)*q(z)}dxdz \\
&= E_x[\int p(z|x)log\frac{p(z|x)*p(x)}{q(x|z)*q(z)}dz] \\
&= E_x\left[\int p(z|x)log\frac{p(z|x)}{q(x|z)*q(z)}dz\right] + E_x[\int p(z|x)logp(x)\,dz] \\
&= E_x\left[\int p(z|x)log\frac{p(z|x)}{q(x|z)*q(z)}dz\right] + E_x[\int p(z|x)dzlogp(x)] \\
&= E_x\left[\int p(z|x)log\frac{p(z|x)}{q(x|z)*q(z)}dz\right] + 常数
\end{aligned}
$$

两个未知的部分，p(z|x)和q(x|z)，使用EM算法，先固定一个求另一个，E步是固定q(x|z)，求期望

$$
\begin{aligned}
&min_{p(z|x)}E_x\left[\int p(z|x)\log\frac{p(z|x)}{q(z|x)*q(x)}dz\right] \\
&= \min_{p(z|x)}E_x\left[\int p(z|x)log\frac{p(z|x)}{q(z|x)}dz - \int p(z|x)dzlogq(x)\right] \\
&= min_{p(z|x)}E_x\left[\int p(z|x)log\frac{p(z|x)}{q(z|x)}dz\right] - 常数 \\
&= \min_{p(z|x)}E_x[KL(p(z|x)\,\|\,q(z|x))]
\end{aligned}
$$

因为这里对p和q都没有限制，所以只要KL直接等于0即可，也就是 p(z|x) = q(z|x) = $\frac{q(x,z)}{\int q(x,z)dz}$。

EM算法通常应用于我们清楚知道原分布形式，但是不知道具体参数的情形，所以我们定义的变分函数q(x, z)，q(z|x)都是很明确的某个分布的概率密度函数，都是好算的，所以在固定q(迭代第一轮时给q的未知参数随机赋值)的时候，后验p(z|x)是能直接算出来的。计算了在前一轮q(x|z)值下p(z|x)的最优解之后，就可以把解出来的p(z|x)代回去原式计算了，即有M步，求期望最大值

$$
\begin{aligned}
q(x|z) = \min_{q(x|z)} KL(p(x,z)\,\|\,q(x,z)) &= \min_{q(x|z)} -E_x[\int p(z|x)logq(x|z)*q(z)] \\
&= \max_{q(x|z)} E_x[\int p(z|x)logq(x,z)]
\end{aligned}
$$

如果用概率图来表示，就是

z生成x：M步，q(x|z)
x推断z：E步，p(z|x)

c. 回顾VAE

当后验不好求的时候，不知道具体分布形式是啥的时候，EM算法就没法用了，这时候就用一个神经网络来模拟分布好了。推导依旧从KL散度开始，直接从上面推导的最后一步开始算

$$\min \mathrm{KL}(p(x,z) \parallel q(x,z)) = minE_x\left[\int p(z|x)log\frac{p(z|x)}{q(x|z)*q(z)}dz\right] + 常数$$
$$= \min E_x\left[-\int p(z|x)logq(x|z)dz + KL(p(z|x) \parallel q(z))\right]$$

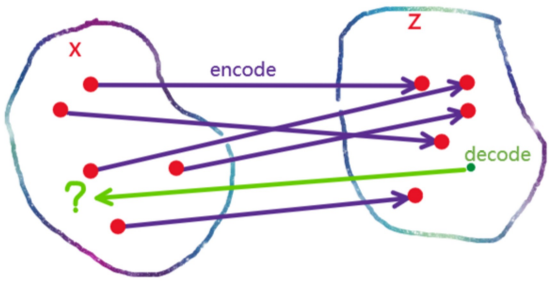
p(z|x)是encoder，q(x|z)是decoder，两个都是神经网络，并且我们预先规定q(z)是标准高斯分布，因此p(z|x)也应该是高斯分布，encoder输出均值和方差，这时候第二项KL散度就可以显式地算出来。第一项积分通过重参数采样一个点可以算出来，因此VAE的最终目标就是

$$\min E_x[-\log q(x|z) + KL(p(z|x) \parallel q(z))]$$

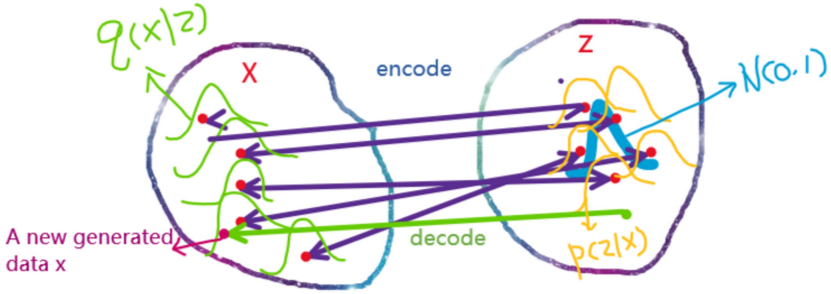
概率图和上图一样，z生成x，即decoder q(x|z)，x推断z，即encoder p(z|x)

VAE假设q(x|z)，p(z|x)和q(z)都是高斯分布，并且优化目标中要求隐变量的后验p(z|x)要与标准高斯分布q(z)的距离尽量近。可能会觉得这样很奇怪，前面也提到过理解VAE的一个误解：以为既然所有这些分布都是高斯的，那么数据x的分布也是高斯的。现在可以尝试再来解释一下(主要是我自己写到这的时候又陷入了迷思，解释给自己听的)，不是严格的理解，就是个人感觉这样理解比较简单。

首先理解下普通的AE。普通的AE就是从x空间到z空间的映射，从x空间中的一个点映射到z空间中的一个点，经过encoder之后，我们得到的是在z空间中的一堆对应的点。但是数据量不会是无限的，所以这些点肯定不会铺满z空间，如果我们直接从z空间中随机采样一个点，这个点很有可能是之前没见过的点，这时候想要通过decoder解码出来一个有用的数据那就是不可能的，因此普通的AE是没有生成功能的，就像下图所示



那要怎么办呢，于是就有人想到了既然点对点encode不行，那把它encode成一个分布怎么样呢？分布可以覆盖一定的范围，这样从z空间采样一个点出来也就可以被包括在里面了。那应该映射成什么分布呢？"世间万物大多可以用高斯分布来描述"，于是就有了假设：p(z|x)是高斯分布。对于每一个数据点x，都映射成一个z空间中的高斯分布，记为$p_i(z_i|x_i)$。光是高斯分布不行啊，高斯分布那么多，怎么限制是哪个呢，而且从z空间采样到底是从哪里采样呢，这个分布应该是简单且容易采样的，于是结合q_i是高斯分布，对于q(z)就选择了标准高斯分布$\mathcal{N}(0,\mathrm{I})$，并且希望每一个$x_i$所属的高斯分布$p_i$都尽量向$\mathcal{N}(0,\mathrm{I})$靠近，这样生成的时候就可以直接从简单分布$\mathcal{N}(0,\mathrm{I})$中采样。既然从x空间到z空间的映射是映射到一个分布，那么decode的时候从z空间到x空间的也应该是一个分布，于是同样地，从z到x是生成一个分布q(x|z)，并且同样选择了高斯分布，这样也能生成一些训练数据以外的样本，而不仅仅是重现原训练数据，更好地实现生成功能。



从上图可以看到，x的后验q(x|z)是高斯分布，是给定了某个z后的分布，而x本身依然是一个很复杂的分布。

d. 用变分推断来分析GAN

轮到GAN这里就比较棘手了，因为现在x是不知道的，z才是能看到的变量，如果仿照前两个一样的分析，将x和z的位置对调，z推断x，x生成z，就会变成

$$\min KL(q(x,z) \| p(x,z)) \Rightarrow \min E_z\left[-\int p(x|z)\log q(z|x)dx + KL(q(x|z) \| p(x))\right]$$

上式当中有好几个分布我们都没办法用GAN的网络来表示，比如p(x)，在前面的情况中，隐变量的先验是预设的，一般为了简单起见都会设常见易求的分布，但是在GAN中，因为x就是实际的数据，不是一个普通的隐变量，先验必定是一个很复杂的分布，无法预设；再比如第一项对x的积分和q(z|x)，在GAN的架构里也是没有可行的解决的。所以对于GAN的变分推断分析，我们需要换一个方向来思考。

GAN中，生成器G是从输入z生成一个x，是z到x的变换，而不是像VAE中的decoder或者encoder都是表示分布 (decoder中的q(x|z)是一个很窄的高斯分布，所以看起来就像它也是直接从z生成x)，因此在GAN中x和z就是一一对应的关系而不是随机关系了，于是也就不存在两者的联合分布p(x, z)之说了。

那么要从变分推断来理解GAN的话，隐变量到底是啥呀，下面介绍从网上看到的两种说法。

首先，我们前面的分析一直忽略了GAN中的判别器discriminator，现在要把它考虑进来了。D以x为输入，输出真或者假的概率，我们将真假标签记为y，y=1即为真，y=0即为假。

i. 第一种说法：y为隐变量

这种说法认为，GAN的判别器就是一个从真实样本或者生成样本x推断y标签的过程，因此将y作为隐变量。y的分布我们是知道的，因为在训练判别器的过程中，每次都会输入一半真实样本和一半生成样本，因此y就是一个二元概率分布，取0和1的概率都是1/2，为了简洁，把y取1的概率记为p_1, 取0的概率记为p_0

$$q(y) = \begin{cases} p_1 = \frac{1}{2}, & y = 1 \\ p_0 = \frac{1}{2}, & y = 0 \end{cases}$$

我们也可以很轻松地写出x基于y的条件分布q(x|y)

$$q(x|y) = \begin{cases} p(x), & y = 1 \\ q(x), & y = 0 \end{cases}$$

其中p(x)是数据真实分布，q(x)是生成样本的分布。因此，x和y的联合分布就可以写成

$$q(x, y) = q(x|y)*q(y) = \begin{cases} p(x)p_1, & y = 1 \\ q(x)p_0, & y = 0 \end{cases}$$

按照前面的分析套路，还是从KL散度开始

$KL(q(x, y) \| p(x, y)) = \iint q(x,y)log\frac{q(x,y)}{p(x,y)}dxdy$ (因为y是离散的取值，积分变成求和)

$$= \sum_y \int q(x,y)log\frac{q(x,y)}{p(x,y)}dx$$

$$= \int q(x, y = 1)log\frac{q(x,y=1)}{p(x,y=1)}dx + \int q(x, y = 0)log\frac{q(x,y=0)}{p(x,y=0)}dx$$

$$= \int p(x)p_1 log\frac{p(x)p_1}{p(x,y=1)}dx + \int q(x)p_0 log\frac{q(x)p_0}{p(x,y=0)}dx$$

$$= \frac{1}{2} * \left[\int p(x) log \frac{1}{p(y=1|x)} dx + \int q(x) log \frac{q(x)}{p(y=0|x)p(x)} dx\right]$$

$\min_{q,p} KL(q(x, y) \| p(x, y)) \sim min \int p(x) log \frac{1}{p(y=1|x)} dx + \int q(x) log \frac{q(x)}{p(y=0|x)p(x)} dx$

来对应GAN的结构看看现在式子里的函数都是什么。p(x)，真实数据分布；q(x)，生成的数据分布，也即网络中的G(z)；p(y=1|x)，给定一个x，求是真实数据的概率，就是网络中的D(x)，p(y=0|x)即1-D(x)。所以，除了真实数据分布以外，现在要优化的式子里有两个未知，q(x)和p(y|x)，正好对应GAN中的G(z)和D(x)。上面的目标函数也就变成了

$$\min KL(q(x, y) \| p(x, y)) \sim \min E_{x\sim p(x)}\left[\log \frac{1}{D(x)}\right] + E_{x\sim q(x)}\left[\log \frac{q(x)}{p(y=0|x)p(x)}\right]$$

要解含有两个未知的目标函数，我们可以运用类似EM算法的思路，固定其中一个，求另一个。先来固定q(x)，求解p(y|x)，也即D(x)，上式中第二项含有的q(x)和p(x)的部分就可以看作常量(q(x)固定了，p(x)是真实数据分布，本来就是不变的常量)

$$E_{x\sim q(x)}\left[\log \frac{q(x)}{p(y=0|x) * p(x)}\right] = E_{x\sim q(x)}[\log q(x)] - E_{x\sim q(x)}[log p(y=0|x) * p(x)] \sim - E_{x\sim q(x)}[\log(1 - D(x))]$$

第一步，固定q(x)，也即G，求解p(y|x)，也即D的优化目标就变成了

$$D = \arg min_D - E_{x\sim p(x)}[log D(x)] - E_{x\sim q(x)}[\log(1 - D(x))] = \arg max_D E_{x\sim p(x)}[log D(x)] + E_{x\sim q(x)}[\log(1 - D(x))]$$

正和前面介绍GAN一节中判别器的目标一模一样！D网络如果有足够的拟合能力，那么目标函数就能取到基于当前固定的q(x)的最优解$D^*(x) = \frac{p(x)}{p(x)+q(x)}$。

我们把当前这个q(x)记作$q^0(x)$，那么当前的D最优解就是$D^*(x) = \frac{p(x)}{p(x)+q^0(x)}$

第二步，固定p(y|x)， 求解q(x)，这时候目标函数中的第一项成了常量，G的优化目标就是

$$G = \arg min_G E_{x\sim q(x)}[\log \frac{q(x)}{(1-D(x))p(x)}]$$

这个目标里含有p(x)，它虽然是常量，但是我们不知道具体形式的p(x)，要设法把它换掉。因为D是固定的，是第一步里面求出来的，从D^*可以求出来

$$D(x) = \frac{p(x)}{p(x)+q^0(x)}$$

$$D(x) * p(x) + D(x) * q^0(x) = p(x)$$

$$(1-D(x))*p(x) = D(x)*q^0(x)$$

把$D(x)*q^0(x)$代到G的优化目标里面去

$$G = \arg min_G E_{x\sim q(x)}[\log \frac{q(x)}{D(x)*q^0(x)}] = argmin_G - E_{x\sim q(x)}[log D(x)] + E_{x\sim q(x)}[log \frac{q(x)}{q^0(x)}] = argmin_G - E_{x\sim q(x)}[log D(x)] + KL(q(x) \| q^0(x))$$

可见第一项是GAN中常用的G的loss，有趣的是比之前的GAN多出来的第二项。第二项是优化后的q(x)和优化前的上一轮求出来的$q^0(x)$的KL距离。也就是说，从变分推断，最小化变分分布和真实分布的KL距离的角度来推导的话，原始GAN的loss其实是一个被低估了的loss，因为还差一个第二项的，与之前的q(x)不能相差太远的限制。

(这个段落的内容我感觉不一定对，选择性看)第二项KL距离是可以估计的。

$$KL(q(x) \| q^0(x)) = KL(q(x|z)q(z) \| q^0(x|z)q(z))$$

q(x|z)就是GAN中生成器所表示的，只是z和x之间不再是随机的分布，而是一一对应的，所以这里我们给它定义一个单点分布，狄拉克分布

$$q(x|z) = \delta(x - G(z)) \begin{cases} \infty, x = G(z) \\ 0, x \neq G(z) \end{cases}$$

$$KL(q(x) \| q^0(x)) = \iint q(x|z)q(z) log \frac{q(x|z)q(z)}{q^0(x|z)q(z)} dxdz = \iint \delta(x - G(z))q(z) log \frac{\delta(x-G(z))q(z)}{\delta(x-G^0(z))q(z)} dxdz$$

$$= \iint \delta(-G(z))q(z)\log\frac{\delta(x-G(z))}{\delta(-G^0(z))}dxdz$$
$$= \int q(z)\log\frac{\delta(0)}{\delta(G(z)-G^0(z))}dz$$

(因为狄拉克函数除了0点以外其他点取值都是0，对x积分就只需计算x=G(z)处的值)

狄拉克分布是个单点分布，也可以用方差趋向于0的高斯分布的极限来表示，即

$$\delta(\mathrm{x}) = \lim_{\sigma\to 0}\frac{1}{(2\pi\sigma^2)^{\mathrm{d}/2}}\exp(-\frac{x^2}{2\sigma^2})$$

将上式代入这个极限的定义，就可以得到

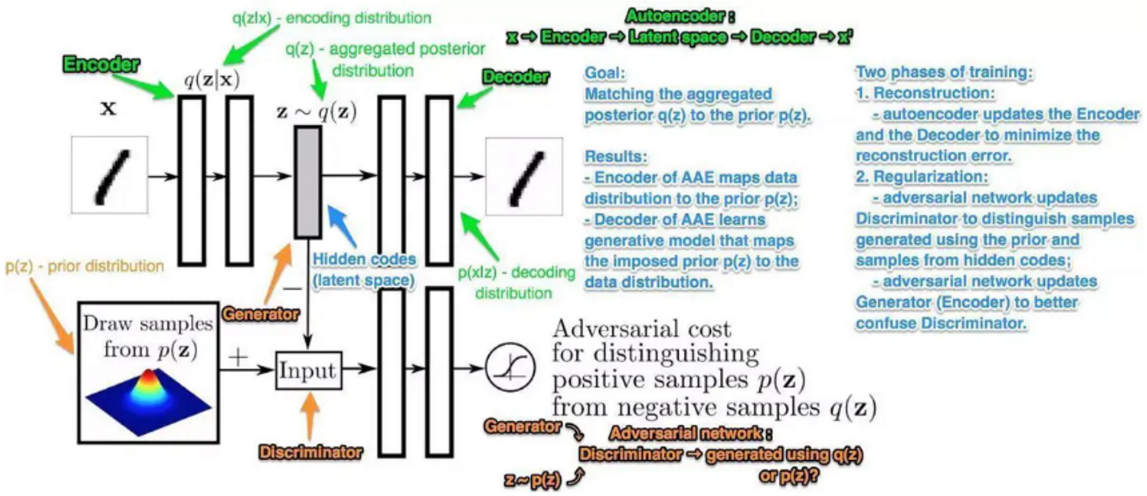
$$\mathrm{KL}(q(x)\parallel q^0(x)) \sim \lambda\int q(z)\|G(z)-G^0(z)\|dz$$
$$= E_{z\sim q(z)}[\|G(z)-G^0(z)\|]$$

所以可以把这一项加到生成器原始的loss中，就可以得到一个更准确的不被低估的loss值了

$$\mathrm{G} = argmin_G - E_{z\sim q(z)}[logD(G(z)) + \|G(z)-G^0(z)\|]$$

上面的推导过程可以与前文的很多地方对应起来。

1) 这里最小化的KL是reverse KL，而非EM和VAE中的正向的KL，而得到的目标函数也正好是GAN中第二种形式的目标函数，因此GAN中第二种形式的loss也是有理论支撑的。
2) EM算法中，变分函数q负责生成，优化目标在M步估计q(x|z)，真实分布负责推断，在E步估计隐变量的后验p(z|x)；VAE算法中，变分函数负责生成，用decoder表示的q(x|z)，真实分布负责推断，用encoder表示的p(z|x)；同样地，GAN也是，我们预设的变分函数q(x|y)负责生成，不过这里的隐变量y是离散的而且我们知道y的分布，所以q(x|y)就可以直接变成了q(x)，也即生成器G(z)生成的样本构成的分布，真实分布负责推断，也即判别器D(x)所表示的p(y|x)。
3) 结合分析过程，会发现GAN和EM很像呀，EM的E步，估计后验p(z|x)，求得似然基于z后验的期望表达式，GAN训练中，会先训练D，也是后验p(y|x)；EM的M步，最大化E步中求得的期望，得到q(x|z)中的未知参数值，而GAN训练生成器G，只是y=1的情况就不需要表示了，只要q(x|y=0)。(下面的第二种说法里会说，以及关于从EM延伸到GAN的理解：https://www.zhihu.com/question/40797593)
4) 在VAE中，选择了q(x|z)是高斯分布，而在GAN中，z和x则是一一对应的关系，且在原始GAN中没有学习从x到z的映射。
5) VAE优化的是正向KL散度，而GAN优化的是reverse KL散度，所以VAE的输出是smoothed的(图像也较为模糊)，GAN的输出是比较清晰的sharp的，但是存在mode collapse问题，因此有不少VAE和GAN结合的模型，希望可以互相取长补短。
    a) Adversarial Autoencoder(AAE)



AAE是将对抗的思想应用到了AE的code学习中。在VAE中，为了让编码z与标准高斯分布靠近，是最小化p(z|x)和q(z)的距离，也即VAE的

loss function中的第二项。而在AAE中，则是运用了对抗的思想，将encoder看作是生成器，给AE额外增加了一个判别器，来判别z是采样自标准高斯分布的还是从encoder中编码出来的。

对AAE的分析只要把对GAN的分析中z和x的位置对调即可。因此就有

D = $\arg max_D E_{z\sim p(z)}[logD(z)] + E_{z\sim q(z)}[log(1-D(z))]$

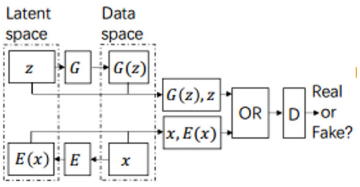
Enc = $argmin_{Enc} - E_{x\sim p(x)}[logDec(Enc(x)) + \lambda \|Enc(x) - Enc^0(x)\|]$

（第二项可以去掉，因为目前不确定前面对于这项分析的正）

Dec = $argmin_{Dec} E_{x\sim p(x)}[\lambda \|x - Dec(Enc(x))\|]$

b) ALI

AAE中是将VAE的AE与GAN的判别器结合，而在ALI中，是将GAN与encoder结合。



(a) BiGAN/ALI

分析也是和GAN一样，只是这时候多了一个z变量，隐变量有z和y，这时变量间的推断关系就是p(x, z, y) = p(y|z, x)*p(z|x)*p(x)，生成关系是q(x, y, z) = $\begin{cases} p(z|x) * p(x) * p_1, & y = 1 \\ q(x|z) * q(z) * p_0, & y = 0 \end{cases}$，仿照GAN的分析，代入简化KL距离即可。

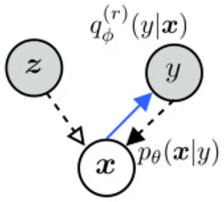
ii. 第二种说法：x为隐变量

在GAN训练过程中，生成的样本x长什么样我们是不知道的，但是我们知道输入的z是什么样的，也知道某个样本的标签是真还是假，所以从这个可见与不可见的角度思考，可以将GAN中生成样本的过程理解为推断x，将判别的过程理解为生成标签y，可见变量就是z和y，隐变量就是x。

在这个设定之下，输入到判别器的数据可以看作是来自两个domain的，一个是real domain，一个是generated domain；而变量z，可以看作与y之间有基于y的条件分布p(z|y)，再由确定的变换函数$G_\theta(z)$，变换到x的基于y的隐式条件分布，这样就没z什么事了，z已经被包含在隐式分布$p_\theta(x|y)$中了

$$x\sim p_\theta(x|y) \Leftrightarrow x = G_\theta(z),\ z\sim p(z|y)$$

变量之间的概率图可以用下图表示



(c)

图示说明：

a) 实线的箭头表示生成过程，虚线箭头表示推断过程。在上图中，x生成y，z和y推断x

b) 从z到x的空心箭头表示一个确定的变换，也即两个变量间是确定对应的关系，这个变换指向隐式分布$p_\theta(x|y)$

c) 实心箭头表示probabilisitic的变换，也即两个变量间是随机分布关系，$p_\theta(x|y)$。图中有从x指向y的实心箭头，表示条件分布q(y|x)；有从y到x的实心箭头，表示分布$p_\theta(x|y)$

d) 蓝色箭头表示对抗，即正向标签y分布$q_\phi(y|x)$和逆向y分布$q_\phi{}^r(y|x)$（x被赋予错误的标签）之间的对抗关系

隐式分布pθ(x|y)的定义和第一种说法是一样的，且y的分布p(y)也是一个二元分布，p(y=1)=p(y=0)=$\frac{1}{2}$

$$\mathrm{p}_\theta(x|y) = \begin{cases} p_\theta(x), & y = 1 \\ q(x), & y = 0 \end{cases}$$

同样地，判别器也用D(x)来表示，记$D_\phi(\mathrm{x}) = q_\phi(\mathrm{y} = 1|\mathrm{x})$，与第一种说法不同的是，这里还定义了逆向的q分布，即标签分类错误的分布，记为$q_\phi{}^r(y|x) = q_\phi(1 - y|x)$。在这样的定义下，GAN的生成器的目标就就可以表示为使得逆向分布的cross entropy loss最小，而判别器目标就是使得正向y分布的cross entropy loss最小

$$\begin{aligned}
\mathrm{G} &= argmin_G - E_{P_\theta(x|y)p(y)}[q_\phi{}^r(y|x)] \\
&= argmax_G E_{P_\theta(x|y)p(y)}[q_\phi{}^r(y|x)] \quad (1) \\
&= \mathrm{argmax}_G E_{p_\theta(x|y=0)p(y=0)}[q_\phi{}^r(y=0|x)] + \\
&E_{p_\theta(x|y=1)p(y=1)}[q_\phi{}^r(y=1|x)] \\
&= \mathrm{argmax}_G E_{p_\theta(x|y=0)p(y=0)}[q_\phi{}^r(y=0|x)] + const \\
&= \tfrac{1}{2} argmax_G E_{x=G_\theta(z),z\sim p(z|y=0)}[q_\phi{}^r(y=0|x)] \\
\mathrm{D} &= argmax_D E_{P_\theta(x|y)p(y)}[q_\phi(y|x)] \quad (2)
\end{aligned}$$

我们来看式子(1)和式子(2)。式子(1)是固定D优化G，即固定ϕ优化θ，式子(2)是固定G优化D，即固定θ优化ϕ，这个思路和第一种说法里的一样的。而且也是EM算法的思路。在EM算法的E步我们计算变量联合分布似然基于后验分布的期望，这里的式子(1)也是计算使得分布q(y|x)似然最优的后验分布，只是比起E步中的联合分布，这里是条件分布，相当于缺少了一个限制后验p(x|y)和q(x)之间距离的正则项。在EM算法的M步是解似然函数里的参数，使得似然最大，这里的(2)式也是一样的。因此在这里面从x生成标签y的生成分布$q_\phi(y|x)$相当于EM中的似然函数，而隐分布pθ(x|y)相当于EM中的后验推断（其实和第一种说法是差不多的，只是描述角度不一样。）

在本文最开始讲解最大似然的时候提过，后验正比于似然乘以先验，也即

$$\begin{aligned}
&\mathrm{q}^r(x|y) \propto q_{\phi_0}{}^r(y|x) * \mathrm{p}_{\theta_0}(x) \\
&p_{\theta_0}(x) = E_{p(y)}[p_{\theta_0}(x|y)] \\
&\quad = \mathrm{p}(\mathrm{y} = 0)p_{\theta_0}(x|y = 0) + p(y = 1)p_{\theta_0}(x|y = 0) \\
&\quad = \mathrm{p}(\mathrm{y} = 0)p_{\theta_0}(x) + p(y = 1)p_{data}(x) \\
&\quad = \tfrac{1}{2}(p_{\theta_0}(x) + p_{aata}(x))
\end{aligned}$$

其中，θ_0和ϕ_0是上一轮迭代中得到的θ值和φ值。这时候我们开始当前轮的迭代，求解关于G，也即θ的优化问题，式子(1)在当前θ处求梯度。可以证明，有

$$\begin{aligned}
&\nabla_\theta \left[-\mathbb{E}_{p_\theta(\mathbf{x}|y)p(y)} \left[\log q^r_{\phi_0}(y|\mathbf{x}) \right] \right] \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0} = \\
&\nabla_\theta \left[\mathbb{E}_{p(y)} \left[KL\left(p_\theta(\mathbf{x}|y) \| q^r(\mathbf{x}|y)\right) \right] - JSD\left(p_\theta(\mathbf{x}|y=0) \| p_\theta(\mathbf{x}|y=1)\right) \right] \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0},
\end{aligned}$$

从这个式子中，我们可以有以下的观察和结论：

a) 上式的第一项，$q^r(x|y)$是真实的后验，pθ(x|y)是需要优化的变分分布，最小化两者之间的KL距离正好就是变分推断的内容

b) 上式的第二项，这一项目标与第一项相反，要求真实分布和生成分布的距离尽量大，可以证明，KL项是JSD项的一个上界，因此优化第一项，最小化KL也就让JSD的范围变小了

c) 优化第一项时，y=1的情况不包含优化目标θ，因此可以作为常量，只剩下y=0的情况

$$\mathrm{KL}\left(p_\theta(\mathbf{x}|y=0) \| q^r(\mathbf{x}|y=0)\right) = \mathrm{KL}\left(p_{g_\theta}(\mathbf{x}) \| q^r(\mathbf{x}|y=0)\right)$$

因为当前的真实后验$q^r(x|y)$是由上一轮的y后验和x分布构成的，而从pθ的定义来看它是一个$p_{\theta 0}(x)$和$p_{data}(x)$的混合分布，因此$q^r(x|y)$也是一个两者的混合。所以优化KL距离，也就是让G往两者的混合分布靠近。（与第一种说法类似）

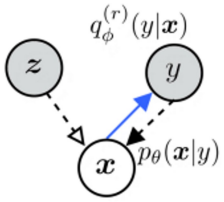
d) 关于mode缺失

KL项中变分分布pθ(x|y)在前面，真实分布$q^r(x|y)$在后面，所以这是一个reverse KL项。（看，我们又从另外一个角度推出来了GAN确实在优化

reverse KL项）

加下来我们用前面的概率图形式来描述其他几个模型，看看它们和GAN的关系。

1）GAN的概率图



(c)

图示说明：

a）可见变量是z和y，隐变量是x

b）实线的箭头表示生成过程，虚线箭头表示推断过程。在上图中，x生成y，z和y推断x

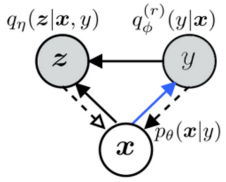
c）从z到x的空心箭头表示一个确定的变换，也即两个变量间是确定对应的关系，这个变换指向隐式分布$p_\theta(x|y)$

d）实心箭头表示probabilisitic的变换，也即两个变量间是随机分布关系，$p_\theta(x|y)$。图中有从x指向y的实心箭头，表示条件分布q(y|x)；有从y到x的实心箭头，表示分布$p_\theta(x|y)$

e）蓝色箭头表示对抗，即正向标签y分布$q_\phi(y|x)$和逆向y分布$q_\phi{}^r(y|x)$（x被赋予错误的标签）之间的对抗关系

2）infoGAN的概率图

infoGAN是在GAN的基础上，增加了要求从x推出生成x的随机变量z，因此概率图只需要比GAN增加从x和y推断出z的关系即可



图示说明：

a）可见变量是z和y，隐变量是x

b）实线的箭头表示生成过程，虚线箭头表示推断过程。在上图中，x生成y，x和y生成z，z和y推断x

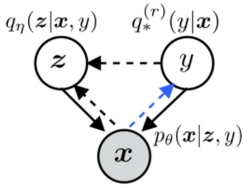
c）从z到x的空心箭头表示一个确定的变换，这个变换指向隐式分布$p_\theta(x|y)$，z和x因为是一一对应关系，已经被包括在这个隐分布里了

d）实心箭头表示probabilisitic的变换，图中的随机关系包括隐式分布$p_\theta(x|y)$，显式分布$q_\phi(y|x)$，$q_\eta(z|x, y)$

e）蓝色箭头表示对抗，即正向标签y分布$q_\phi(y|x)$和逆向y分布$q_\phi{}^r(y|x)$（x被赋予错误的标签）之间的对抗关系

3）VAE的概率图

VAE与GAN刚好对称，GAN是随机变量z和真假两个domain隐式地推断出x的分布，而VAE是从数据x推断出z。VAE中只有真实数据的样本，所以当中q(y|x)的对抗关系是退化了的，VAE可以看作GAN的特殊情况。VAE的概率图也是和infoGAN相反，在infoGAN中变量间的推断关系在VAE中是生成关系，在infoGAN中的生成关系在VAE中是推断关系。

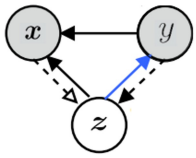


图示说明：

a) 可见变量是x，隐变量是z和y
b) 实线的箭头表示生成过程，虚线箭头表示推断过程。在上图中，z和y生成x，x推断y，x和y推断z
c) VAE显式地学习数据分布，没有隐分布，变量间关系都是随机分布的关系
d) 实心箭头表示probabilisitic的变换，图中的随机关系包括p_θ(x|z, y)，q_ϕ(y|x)，q_η(z|x, y)
   i) 由于在VAE中y是退化了的，所以没有了对抗关系，当x是真实数据时，p(y=1|x)恒等于1
   ii) $p_\theta(x|z,y)=\begin{cases}p_\theta(x|z), & y=0, \text{ decoder}\\ p_{data}(x), & y=1\end{cases}$
   iii) 对于q_η(z|x, y)，q_η(z|x, y=1)是常量，q_η(z|x, y=0)即encoder所表示的q_η(z|x)
e) 蓝色箭头表示对抗，即正向标签y分布q_ϕ(y|x)和逆向y分布${q_\phi}^r(y|x)$（x被赋予错误的标签）之间的对抗关系，但是这个关系是退化了的，VAE中并没体现出来

4) AAE的概率图

在第一种说法中已经介绍过AAE了，就是在普通AE的基础上，将encoder的输出z与从标准高斯分布里采样到的z放到一个判别器里面去进行对抗学习。下面的概率图与infoGAN的概率图类似，但是x和z的位置对调了。infoGAN是推断的x和真实x拿去对抗，并且要生成出z；而AAE是推断的z和标准高斯z拿去对抗，并生成出x，两者是一种对称的关系。



(c)

图示说明：

a) 可见变量是x和y，隐变量是z
b) 实线的箭头表示生成过程，虚线箭头表示推断过程。在上图中，z和y生成x，z生成y，x和y推断z
c) 实心箭头表示probabilisitic的变换。
d) 蓝色箭头表示对抗，即正向标签y分布q_ϕ(y|z)和逆向y分布${q_\phi}^r(y|z)$（z被赋予错误的标签）之间的对抗关系

5) CycleGAN（把infoGAN和AAE两个图的线结合起来）

cycleGAN是两个domain之间的相互变换，它有两个生成器，负责两个方向的变换，两个判别器负责两个域的判别。

cycleGAN可以看作是AAE和infoGAN的结合。AAE的目标可以看作从x变换到z，infoGAN则是从z变换到x。

关于wake sleep算法（一种广义的EM算法）：

GAN和VAE的关系与wake sleep算法中wake阶段和sleep阶段的关系类似。Wake阶段对应EM的M，最大化基于后验分布的似然期望；Sleep阶段对应EM的E，优化隐变量的后验分布

wake：醒着时可以看见真实数据，优化似然p(x|h)，相当于VAE的decoder，VAE扩展了wake阶段，通过指定z先验分布学习了推断模型

sleep：睡着时看不见真实数据，从生成数据优化推断q(h|x)，相当于GAN的discriminator，GAN扩展了sleep阶段，学习了生成模型

e. 总结

本文详细介绍的三种算法：EM, VAE和GAN，都可以归纳到变分推断的框架下。EM是后验分布简单易算的变分推断，VAE在优化变分推断中的ELBO的基础上，额外增加了隐

变量先验的限定并要求后验与先验尽量靠近，而GAN则是在变分推断的基础上，加入了对抗思想，一方面变分分布要与真实分布靠近，而另一方面，变分分布每次更新的变化又不能太大，最终就是变分分布每次都往两个分布的混合分布靠近。普遍观点认为，VAE优化正向KL距离，因此生成样本比较平滑，是数据的mixture，以图片生成为例具体表现就是图片比较模糊，而GAN优化逆向KL距离，倾向于选择较突出的mode，以图片生成为例具体表现就是图片比较清晰，但是存在mode collapse问题。两者是互相对称的。

其实VAE和GAN都是在处理隐空间z和数据空间x之间的分布映射问题，只是处理方式不一样。

最后：在motion synthesis上没有模糊一说，所以VAE和GAN的效果区别有待实验研究，在实验后再补充。。。