**自动编码变化贝叶斯**

**D**iederik P. Kingma机器学习小组阿姆斯特丹大学[dpgang@ gmail](mailto:dpkingma@gmail.com)

**马**克斯韦林机器学习小组阿姆斯特丹大学

[Weln.Max @ Gmail](mailto:welling.max@gmail.com)

# 摘要

arXiv：1312.6114v10 [stat.ML] 2014年5月1日

在存在具有难解的后验分布的连续潜在变量和大型数据集的情况下，我们如何在定向概率模型中进行有效的推理和学习？我们介绍了一种随机变分推理和学习算法，该算法可扩展到大型数据集，并且在某些温和的微分条件下，甚至可以在难处理的情况下工作。我们的贡献是双重的。首先，我们证明了变化下界的重新参数化产生了一个下界估计量，该估计量可以使用标准随机梯度方法直接优化。其次，我们表明，对于每个数据点具有连续潜在变量的iid数据集，通过使用拟议的下界估计器将近似推理模型（也称为识别模型）拟合到难处理的后验，可以使后验特别有效。理论上的优势体现在实验结果上。

# 介绍

我们如何使用连续的潜在变量和/或参数具有难以处理的后验分布的定向概率模型进行有效的近似推理和学习？变分贝叶斯（VB）方法涉及对难治性后牙的近似优化。不幸的是，普通的均值场方法需要对期望值进行近似后验的解析解，这在一般情况下也是很难解决的。我们展示了变分下界的重新参数化如何产生下界的一个简单的可微分的无偏估计量。该SGVB（随机梯度变化贝叶斯）估计器可用于几乎任何具有连续潜在变量和/或参数的模型中的有效近似后验推断，并且可以使用标准随机梯度上升技术直接进行优化。

对于iid数据集和每个数据点连续的潜在变量的情况，我们提出了自动编码VB（AEVB）算法。在AEVB算法中，我们通过使用SGVB估计器优化识别模型来使推理和学习特别有效，该模型使我们能够使用简单的祖先采样执行非常有效的近似后验推理，从而使我们能够高效地学习模型参数，而无需每个数据点需要昂贵的迭代推理方案（例如MCMC）。所学习的近似后验推理模型也可以用于许多任务，例如识别，去噪，表示和可视化目的。当将神经网络用于识别模型时，我们得出了变分自动编码器。

# 方法

本节中的策略可用于为具有连续潜在变量的各种有向图模型推导下界估计量（随机目标函数）。在这里，我们将自己限制为常见情况，在这种情况下，我们有一个iid数据集，每个数据点都有潜在变量，并且我们希望对（全局）参数执行最大似然（ML）或最大后验（MAP）推断，并采用变分推断在潜在变量上。例如，

*φ* **z** *θ*

## x

*N*

图1：正在考虑的有向图模型的类型。实线表示生成模型pθ（z）pθ（x | z），虚线表示对难处理的后部pθ（z | x）的变化近似qφ（z | x）。与生成模型pa-一起学习变化参数φ

参数θ。

直接将此场景扩展到我们还对全局参数执行变分推断的情况；该算法被放在附录中，但是这种情况下的实验留待以后的工作。请注意，我们的方法可以应用于在线非固定设置，例如流数据，但是为了简单起见，这里我们假设使用固定的数据集。

## 问题场景

让我们考虑一些数据集X = {x(i) N，其中包含一些连续或离散变量x的N iid个样本。我们假设数据是通过某种随机过程生成的，涉及一个未观察到的连续随机变量z。该过程包括两个步骤：（1）从某个先验分布pθ∗（z）生成值z(i)；（2）从某些条件生成一个值x(i)-

*}i*=1

有理分布pθ∗（x | z）。我们假设先验pθ∗（z）和似然性pθ∗（x | z）来自分布pθ（z）和pθ（x | z）的参数族，并且它们的PDF几乎可以区分

θ和z都无处不在。不幸的是，我们看不到很多这样的过程：真正的参数θ*∗*以及潜在变量z(i)的值对我们来说是未知的。

非常重要的是，我们不对边际概率或后验概率进行通用的简化假设。相反，我们对通用算法感兴趣，该通用算法在以下情况下甚至可以高效工作：

* + 1. *难*处理性：边际可能性pθ（x）的积分=

(*p*θ（z）pθ（x | z）dz很难处理（因此我们无法评估或区分边缘可能性），其中真实后验密度pθ（z | x）=pθ（x | z）pθ（z） /pθ（x）很难处理

（因此无法使用EM算法），并且任何合理的均值场VB算法所需的积分也是难解的。这些难点很普遍

并出现在中等复杂的似然函数pθ（x | z）的情况下，例如神经

具有非线性隐藏层的网络。

* + 1. *庞*大的数据集：我们有太多的数据，以至于批处理优化的成本太高；我们希望使用小型迷你批次甚至单个数据点进行参数更新。通常，基于采样的解决方案（例如蒙特卡洛EM）太慢，因为它涉及每个数据点通常昂贵的采样循环。

我们对上述情况下的三个相关问题感兴趣，并提出了解决方案：

1. 对参数θ的有效近似ML或MAP估计。参数本身可能很有趣，例如，如果我们正在分析某个自然过程。它们还使我们能够模仿隐藏的随机过程，并生成类似于真实数据的人工数据。
2. 给定观测值x的潜在变量z的有效近似后验推断

用于选择参数θ。这对于编码或数据表示任务很有用。

1. 变量x的有效近似边际推断。这使我们能够执行所有需要先于x的推理任务。计算机视觉的常见应用包括图像去噪，修复和超分辨率。

为了解决上述问题，让我们介绍一个识别模型qφ（z | x）：近似于难处理的真实后验pθ（z | x）。请注意，与近似

在平均场变分推论的后面，它不一定是阶乘的，并且它的参数φ并不是从某种封闭形式的期望中计算出来的。相反，我们将介绍一种与生成模型参数θ一起学习识别模型参数的方法。

从编码理论的角度来看，未观察到的变量z具有潜在表示或代码的解释。因此，在本文中，我们还将参考识别模型qφ（z | x）

作为一个概率编码器，由于给定了一个数据点x，因此它产生了一个分布（例如高斯分布）

代码z可能从中生成数据点x的可能值。同样，我们将pθ（x | z）称为概率解码器，因为给定代码z会产生一个

分布在x的可能对应值上。

## 变分界

边际可能性由各个数据点的边际可能性之和组成

对数pθ（x(1)，···，x(N )）= },N

*i*=1

logpθ（x(i)），可以分别重写为：

log pθ（x(i)）＝dkl（qω（z x x(i)）p pθ（z x x(i)）+l（θ，ω；x(i)） (1)

第一个RHS项是与真实后验近似的KL散度。从此

KL散度是非负的，第二个RHS项L（θ，φ; x(i)）被称为数据点i的边缘似然的（变分）下界，可以写成：

log p th\*（x(i)）超过L（Th，ω；x(i)）＝E*q*

***φ***

也可以写成：

(z|x)〔log q〕

（z | x）+对数pθ

(**x***,* **z**)] (2)

*L*（θ，φ; x(i)）= -DKL（qφ（z | x(i)）||pθ（z））+ E*q*

***φ***

(**z***|***x**(*i*))

对数pθ（x(i)| z）

(3)

我们希望通过变化参数φ和生成参数θ来区分和优化下限L（θ，φ; x(i)）。然而，下限wrtφ的梯度有点问题。这类问题的常用（幼稚）蒙特卡洛梯度估计器

是：∇φE*qω（z）*[f（z）] = E*qω（z）1*f（z）∇*qω（z）*logqφ（z）l“'

1 },

*L*

*L*

*l*=1

*f* (z)∇*φ*

(z（L）)对数qω（z）

(*l*)

）哪里

**z**(l)〜qφ（z | x(i)）。该梯度估算器显示出非常高的方差（请参见例如[[BJP12]](#_bookmark13)

并且对于我们的目的而言是不切实际的。

## SGVB估计器和AEVB算法

在本节中，我们将介绍下界及其参数的导数的实际估计量。我们假设形式为qφ（z | x）的近似后验，但请注意

这种技术也可以应用于qφ（z）的情况，即我们也不以x为条件。附录中给出了用于推断参数后验的全变分贝叶斯方法。在本节概述的某些温和条件下[2.4](#_bookmark5) 对于选定的近似后验qφ（z | x），我们

**z**

可以使用微分变换gφ（E，x）重新参数化随机变量-〜qφ（z | x）

（辅助）噪声变量E：

**z** =gφ（E，x）其中 ***E*** *∼ p*(***E***) (4)

见章节[2.4](#_bookmark5) 选择这种适当分布p（E）和函数的一般策略

-

*g*φ（E，x）。现在，我们可以形成对某些函数f（z）wrt的期望的蒙特卡洛估计

*q*φ（z | x）如下：

E*q（z）x*（i）)\f（z）= = Ep（E）

*f（gω（e，x）*

(*i*)))

*L*

\

1

*"' L*

*l*=1

*f（gω（e）*

(*l*)

*,* **x**(*i*)

））在哪里 ***E***

(*l*)

*∼ p*(***E***) (5)

我们将此技术应用于变分下限（eq。[(2)),](#_bookmark2) 产生我们的随机

梯度变化贝叶斯（SGVB）估计器L*A*

-

*L*

(***θ****,* ***φ***; **x**

(*i*)

) *"' L*(***θ****,* ***φ***; **x**

(*i*)):

*LA*(θ, φ; x

-

(*i*)

) = 1 \

*L*

*l*=1

对数pθ（x

(*i*)

*,* **z**(*i,l*)

）-对数qφ（z

(*i,l*)

(*i*)

*|*

**x**

)

哪里 **z**(i,l)*=gφ（E*(i,l)*，x*(i)*）和* ***E***(*l*) *∼ p*(***E***) (6)

**算**法1自动编码VB（AEVB）算法的最小批量版本。本节中的两个SGVB估计器之一[2.3](#_bookmark3) 可以使用。在实验中，我们使用设置M = 100和L = 1。

***θ***，φ←初始化参数

## 重复

**X***M*←M个数据点的随机小批量（从完整数据集中得出）

***E***←噪声分布p（E）中的随机样本

**G***~（γ）Th，ωLM*

-

(θ, φ; X*M*

*，*E）（小批量估计量的梯度[(8))](#_bookmark4)

***θ***，φ←使用梯度g更新参数（例如SGD或Adagrad[[DHS10])](#_bookmark16)

**直**到参数（θ，φ）收敛

**返回θ，φ**

通常，等式的KL发散DKL（qφ（z | x(i)）||pθ（z））等于。[(3)](#_bookmark2) 可以分析整合（请参阅

附录[B),](#_bookmark32) 这样只有预期的重建误差E*q（z）x*（i）) 1logpθ（x

(*i*)

*|* z）l要求

通过抽样估算。然后，KL散度项可以解释为正则化φ，

将近似后部老化，使其接近先前的pθ（z）。这产生了第二版

SGVB估算器L*B*

-

(***θ****,* ***φ***; **x**

(*i*)

) *"' L*(***θ****,* ***φ***; **x**

(*i*)

），对应于等式。[(3),](#_bookmark2) 通常较少

比一般估算器的方差：

*L*

*LB*(θ, φ; x

-

(*i*)

＝dkl（qω）（z×x）

(*i*)

)*||p* (**z**)) + 1 \

*l*=1

***θ*** *L*

（对数pθ（x

(*i*)

(*i,l*)

*|*

**z**

))

哪里 **z**(i,l)*=gφ（E*(i,l)*，x*(i)*）和* ***E***(*l*) *∼ p*(***E***) (7)

给定来自具有N个数据点的数据集X的多个数据点，我们可以基于最小批处理构造整个数据集的边际似然下限的估计量：

*M*

*L*(***θ****,* ***φ***; **X**) *"' L*-

*M*

(θ, φ; X*M*

) = *N* \

*M*

*i*=1

*L*-(***θ****,* ***φ***; **x**

(*i*)

) (8)

其中小批量X*M*= {x(i)}*M*

*i*=1

是随机抽取的M个数据点的样本

具有N个数据点的完整数据集X。在我们的实验中，我们发现样本数L

只要小批量大小M足够大（例如M = 100），就可以将每个数据点设置为1。

可以取导数∇θ，φL-（θ; X*M*），所得的梯度可以与SGD或Adagrad等随机优化方法结合使用[[dhs10]。](#_bookmark16) 查看算法[1](#_bookmark0) 为一个

计算随机梯度的基本方法。

当查看eq给出的目标函数时，与自动编码器的连接将变得清晰。[(7).](#_bookmark4) 第一项是（与先验值近似后验的KL散度）充当正则项，而第二项是预期的负重构误差。选择函数gφ（。），以便将数据点x(i)和随机噪声向量E(l)映射到样本中的样本。

该数据点的近似后验：z(i,l)=gφ（E(l)，x(i)）其中z(i,l)〜qφ（z | x(i)）。随后，将样本z(i,l)输入到函数logpθ（x(i)| z(i,l)），该概率等于概率

给定z(i,l)，在生成模型下数据点x(i)的密度（或质量）。这个名词是负数

*自*动编码器说法中的重构错误。

## 重新参数化技巧

为了解决我们的问题，我们调用了一种从qφ（z | x）生成样本的替代方法。基本的参数设置技巧非常简单。令z为连续随机变量，z〜qφ（z | x）为条件分布。然后通常可以表达

随机变量z作为确定性变量z =gφ（E，x），其中E是具有独立边际p（E）的辅助变量，而gφ（。）是由φ参数化的某些矢量值函数。

该重新参数化对我们的情况很有用，因为它可以用来重写期望值wrtqφ（z | x），以使期望值的蒙特卡罗估计可区分为wrtφ。证明如下。给定确定性映射z =gφ（E，x），我们知道qφ（z | x） 你dzi = p（E） 你dEi。因此[1](#_bookmark6)， (qφ（z | x）f（z）dz = (p（E）f（z）dE = (p（E）f（gφ（E，x））dE。它跟随

1请注意，对于无穷小，我们使用符号约定dz = ndzi

*i*

可以构造一个可微估计器： (qφ（z | x）f（z）dz“' 1 },L

*f*（gω（x，e(l)））

*L l*=1

式中E(l)〜p（E）。在部分[2.3](#_bookmark3) 我们运用了这一技巧来获得

变异下限。

以单变量高斯情况为例：令z〜p（z | x）= N（µ，σ2）。在这种情况下，有效的重新参数化为z = µ +σE，其中E是辅助噪声变量E〜N（0，1）。因此，

1 *L* (*l*)

(*l*)

E*N (z;µ,σ*2 )[f（z）] = E*N (E;0,1)*[f（µ +σE）]'' *L },l=1*f（µ +σE

）其中E

*∼ N* (0*,* 1).

我们可以为qφ（z | x）选择这样的微分变换gφ（。）和辅助变量

***E*** ∼ p(E)?三种基本方法是：

* + 1. 可逆CDF。在这种情况下，令E〜U（0，I），并且令gφ（E，x）为的逆CDF。

*q*φ（z | x）。示例：指数，柯西，逻辑，瑞利，帕累托，威布尔，倒数，

Gompertz，Gumbel和Erlang分布。

* + 1. 类似于高斯示例，对于任何“位置比例”分布族，我们都可以选择标准分布（位置= 0，比例= 1）作为辅助变量

***E***，然后让g（。）=位置+比例尺·E。例如：拉普拉斯，椭圆形，学生的t，逻辑

均匀分布，三角形分布和高斯分布。

* + 1. 组成：通常可以将随机变量表示为辅助变量的不同变换。示例：对数正态（正态分布变量的幂），伽玛（指数分布变量的总和），狄利克雷特（伽玛变量的加权总和），贝塔，卡方和F分布。

当所有三种方法都失败时，存在与逆CDF的良好近似，需要时间复杂度可与PDF相比的计算（请参见例如[[Dev86]](#_bookmark15) 对于某些方法）。

# 示例：变体自动编码器

在本节中，我们将给出一个示例，其中我们将神经网络用于概率编码器

*q*φ（z | x）（生成模型pθ（x，z）的后验近似值），其中

φ和θ与AEVB算法共同优化。

令先验的潜在变量为中心各向同性多元高斯pθ（z）=

*N* (z; 0, I).请注意在这种情况下，先验缺少参数。我们让pθ（x | z）是一个多元变量

高斯（对于实值数据而言）或伯努利（对于二进制数据而言），其分布为

参数是使用MLP（具有单个隐藏层的全连接神经网络，从z计算的，请参见附录）[C).](#_bookmark33) 请注意，在这种情况下，真实的后验pθ（z | x）很难处理。尽管形式为qφ（z | x）的自由度很大，但我们假设真实的（但难处理的）后验近似

具有近似对角协方差的高斯形式。在这种情况下，我们可以让变分近似后验为具有对角协方差结构的多元高斯[2](#_bookmark8):

对数qω（Z\* x(i)）＝log n（z；(i)， 9）i） (9)

其中近似后验的均值和sd µ(i)和σ(i)是编码MLP的输出，即数据点x(i)的非线性函数和变分参数φ（请参见附录）[C).](#_bookmark33) 如本节所述[2.4,](#_bookmark5) 我们使用z(i,l)=从后z(i,l)〜qφ（z | x(i)）进行采样

*g*φ（x(i)，E(l)）=μ(i)+σ(i)8 E(l)其中E(l)〜N（0，I）。用8表示元素乘积。在该模型中，pθ（z）（先验）和qφ（z | x）均为高斯。在这种情况下，我们可以使用

方程的估计量[(7)](#_bookmark4) 无需计算即可计算和区分KL差异的地方（请参阅附录）[B).](#_bookmark32) 此模型和数据点x(i)的最终估计量为：

*J L*

*L*（θ，φ; x(i)）'' 1 \ /1 + log（（σ(i)）2）-（µ(i)）2-（σ(i)）21 + 1 \日志p

(**x**(*i*)*|***z**(*i,l*))

2

*j*=1

*j j j*

*L* ***θ***

*l*=1

其中z(i,l)= µ(i)+σ(i)8 E(l) 和 ***E***(*l*) *∼ N* (0*,* **I**) (10)

如上文和附录中所述[C,](#_bookmark33) 解码项对数pθ（x(i)| z(i,l)）是​​伯努利或高斯MLP，具体取决于我们要建模的数据类型。

2请注意，这只是（简化）选择，而不是我们方法的限制。

# 相关工作

唤醒睡眠算法[[HDFN95]](#_bookmark20) 据我们所知，是文献中唯一适用于同一类连续潜变量模型的其他在线学习方法。像我们的方法一样，唤醒睡眠算法采用近似真实后验的识别模型。唤醒睡眠算法的一个缺点是，它需要同时优化两个目标函数，这两个目标函数不完全对应于边际可能性的优化（一个边界）。唤醒睡眠的一个优点是它也适用于具有离散潜在变量的模型。唤醒睡眠每个数据点的计算复杂度与AEVB相同。

随机变分推论[[HBWP13]](#_bookmark19) 最近受到越来越多的关注。最近，[[BJP12]](#_bookmark13) 引入了控制变量方案以减少本节中讨论的朴素梯度估计量的高方差[2.1,](#_bookmark1) 并应用于后验的指数族近似。在[[RGB13]](#_bookmark23) 为了减少原始梯度估计量的方差，引入了一些通用方法，即控制变量方案。在[[SK13]](#_bookmark26) 在有效版本的随机变分推断算法中，使用了与本文类似的重新参数化方法，以学习指数族近似分布的自然参数。

AEVB算法揭示了定向概率模型（受变分目标训练）与自动编码器之间的联系。线性自动编码器和某一类生成线性高斯模型之间的联系早已为人所知。在[[Ro98]](#_bookmark25) 结果表明，PCA

对应于先验p（z）= N（0，I）和条件分布p（x | z）= N（x; Wz的线性高斯模型特殊情况的最大似然（ML）解，EI），特别是

E无限小的情况

在有关自动编码器的最新工作中[[vll+10]](#_bookmark28) 结果表明，非正规自动编码器的训练准则对应于下限的最大化（请参阅infomax原理[[林89]](#_bookmark22) 输入X和潜在表示Z之间的互信息。最大化（wrt参数）等同于最大化条件熵，该条件熵在自动编码模型下受数据预期对数似然性的限制[[VLL+]](#_bookmark28)10]，即负重构误差。但是，众所周知，这种重建标准本身不足以学习有用的表示形式[[bcv13]。](#_bookmark12) 已经提出了正则化技术以使自动编码器学习有用的表示形式，例如去噪，压缩和稀疏自动编码器变体[[bcv13]。](#_bookmark12) SGVB目标包含由变化范围（例如，等式[(10)),](#_bookmark7) 缺乏学习有用表示法所需的通常的烦扰正则化超参数。相关的还有编解码器架构，例如预测稀疏分解（PSD）[[KRL08]](#_bookmark21) 从中我们得到了一些启发。同样重要的是最近引入的生成随机网络[[BTL13]](#_bookmark14) 嘈杂的自动编码器学习从数据分布中采样的马尔可夫链的过渡算子。在[[SL10]](#_bookmark27) 识别模型用于Deep Boltzmann Machines的高效学习。与我们提出的用于学习一般类型的有向概率模型的算法相比，这些方法针对的是非规范化模型（即像Boltzmann机器这样的无向模型）或限于稀疏编码模型。

最近提出的DARN方法[[GMW13],](#_bookmark18) 还使用自动编码结构学习了定向概率模型，但是它们的方法适用于二进制潜在变量。最近，[[RMW14]](#_bookmark24) 还使用我们在本文中描述的重新参数化技巧，在自动编码器，定向概率模型和随机变异推理之间建立联系。他们的工作独立于我们的工作而开发，并提供了有关AEVB的其他观点。

# 实验

我们训练了来自MNIST和Frey Face数据集的图像生成模型[3](#_bookmark9) 并根据变分下界和估计的边际可能性对学习算法进行了比较。

章节中的生成模型（编码器）和变分近似（解码器）[3](#_bookmark7) 在所描述的编码器和解码器具有相等数量的隐藏单元的情况下，使用了这些单元。由于Frey Face数据是连续的，因此我们使用了具有高斯输出的解码器，该编码器与编码器相同，不同之处在于，在S轴上使用S形激活函数将均值约束为间隔（0，1）。

3可在[http://www.cs.nyu.edu/～roweis/data.html](http://www.cs.nyu.edu/%7Eroweis/data.html)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | MNIST， Nz #3 |  | MNIST， Nz #5 |  | MNIST， Nz #10 |  | MNIST， Nz #20 |  | MNIST， Nz +200 |
| 100 |  | 100 |  | 100 |  | 100 |  | 100 |  |
| 110 |  | 110 |  | 110 |  | 110 |  | 110 |  |
| 120 |  | 120 |  | 120 |  | 120 |  | 120 |  |
| 130 |  | 130 |  | 130 |  | 130 |  | 130 |  |
| 140 |  | 140 |  | 140 |  | 140 |  | 140 |  |

150

L

105 106 107 108

150

105 106 107 108

150

105 106 107 108

150

105 106 107 108

150

105 106 107 108

＃评估训练样本

1600

1400

1200

唤醒睡眠（测试）

唤醒睡眠（火车）

AEVB（测试）

AEVB（火车）

1000

800

L

600

400

200

0

弗雷脸， Nz #2

1600

1400

1200

1000

800

600

400

200

0

弗雷脸， Nz #5

1600

1400

1200

1000

800

600

400

200

0

弗雷脸， Nz #10

1600

1400

1200

1000

800

600

400

200

0

弗雷脸， Nz #20

105 106 107 108

105 106 107 108

105 106 107 108

105 106 107 108

图2：针对不同维度的潜在空间（Nz），在优化下限方面，我们的AEVB方法与唤醒睡眠算法的比较。我们的方法收敛更快，在所有实验中都达到了更好的解决方案。有趣的是，更多的潜在变量不会导致更多的过度拟合，这可以通过下限的正则化效应来解释。垂直轴：每个数据点的估计平均变化下限。估计量方差很小（<1），因此省略了。横轴：评估的训练点数。使用有效运行40 GFLOPS的Intel Xeon CPU，每百万培训样本大约需要20-40分钟的计算时间。

解码器输出。注意，对于隐藏单元，我们指的是编码器和解码器的神经网络的隐藏层。

使用随机梯度上升来更新参数，其中通过微分下界估计量∇θ，φL（θ，φ; X）计算梯度（请参阅算法）[1),](#_bookmark0) 加上对应于先前p（θ）= N（0，I）的小权重衰减项。此目标的优化等效于

配对MAP估算，其中似然梯度由下限的梯度近似。

我们将AEVB的性能与唤醒睡眠算法进行了比较[[hdfn95]。](#_bookmark20) 我们将相同的编码器（也称为识别模型）用于唤醒睡眠算法和变分自动编码器。所有参数，包括变量参数和生成参数，均通过从

*N*（0，0.01），并使用MAP准则随机联合优化。使用Adagrad调整步长[[DHS10];](#_bookmark16) Adagrad全局步长参数是根据前几次迭代中训练集的性能从{0.01，0.02，0.1}中选择的。迷你批

*使*用M = 100，每个数据点L = 1个样本。

**可**能性下限我们训练了生成模型（解码器）和相应的编码器（又称为识别模型）（对于MNIST，具有500个隐藏单元，对于Frey Face数据集，具有200个隐藏单元（以防止过度拟合，因为它是相当小的数据集） ）。隐藏单元的选择数量是基于有关自动编码器的现有文献，并且不同算法的相对性能对这些选择不是很敏感。数字[2](#_bookmark10) 显示比较下限时的结果。有趣的是，多余的潜在变量不会导致过度拟合，这可以通过变分界的正则化性质来解释。

**边**际可能性对于非常低维的潜在空间，可以使用MCMC估计器估计学习的生成模型的边际可能性。附录中提供了有关边际似然估计器的更多信息。对于编码器和解码器，我们再次使用了神经网络，这次具有100个隐藏单元和3个潜在变量。对于更高维的潜在空间，估计变得不可靠。同样，使用了MNIST数据集。将AEVB和Wake-Sleep方法与带有混合Monte Carlo（HMC）的Monte Carlo EM（MCEM）进行了比较[[DKPR88]](#_bookmark17) 采样器细节在附录中。对于小型和大型训练集，我们比较了三种算法的收敛速度。结果如图[3.](#_bookmark11)

100 N培养= 1000

125

N培养= 50000

110

边际对数似然

120

130

140

150

130

135

唤醒睡眠（火车）

唤醒睡眠（测试）MCEM（训练）

MCEM（测试）AEVB（火车）AEVB（测试）

140

145

150

155

160

0 10 20 30 40 50 60

160

0 10 20 30 40 50 60

＃评估的训练样本（百万）

图3：就估计的边际可能性而言，对于不同数量的训练点，AEVB与唤醒睡眠算法和Monte Carlo EM的比较。蒙特卡洛EM不是在线算法，并且（不同于AEVB和唤醒睡眠方法）不能有效地应用于完整的MNIST数据集。

**高**维数据的可视化如果选择低维潜在空间（例如2D），则可以使用学习的编码器（识别模型）将高维数据投影到低维流形。见附录[A](#_bookmark29) 用于可视化MNIST和Frey Face数据集的2D潜在歧管。

# 结论

我们介绍了一种新颖的变分下界估计器，随机梯度VB（SGVB），用于使用连续潜在变量进行有效的近似推断。可以使用标准随机梯度法直接区分和优化建议的估算器。对于iid数据集和每个数据点连续的潜在变量的情况，我们引入了一种有效的推理和学习算法，即自动编码VB（AEVB），该算法使用SGVB估算器来学习近似推理模型。理论上的优势体现在实验结果上。

# 未来的工作

由于SGVB估计器和AEVB算法几乎可以应用于具有连续潜在变量的任何推理和学习问题，因此有很多未来的方向：（i）使用用于编码器的深度神经网络（例如卷积网络）学习分层生成体系结构和解码器，由AEVB联合培训；（ii）时间序列模型（即动态贝叶斯网络）；（iii）将SGVB应用于全局参数；（iv）具有潜在变量的监督模型，对于学习复杂的噪声分布很有用。

# 参考文献

[BCV13] Yoshua Bengio，Aaron Courville和Pascal Vincent。代表性学习：回顾与新观点。2013。

[BJP12] David M Blei，Michael I Jordan和John W Paisley。带有随机搜索的变分贝叶斯推理。在第29届国际机器学习会议（ICML-12）的会议记录中，第1367–1374页，2012年。

[BTL13] Yoshua Bengio和E´ric Thibodeau-Laufer。反向生成器可训练的深度生成随机网络。arXiv预印本arXiv：1306.1091，2013年。

[Dev86] Luc Devroye。基于样本的非均匀随机变量生成。在第18届冬季模拟会议论文集中，第260-265页。ACM，1986年。

[DHS10] John Duchi，Elad Hazan和Yoram Singer。在线学习和随机优化的自适应次梯度方法。机器学习研究杂志，2010年12月：2121-2159。

[DKPR87] Simon Duane，Anthony D Kennedy，Brian J Pendleton和Duncan Roweth。混合蒙特卡洛。物理字母B，195（2）：216-222，1987。

[GMW13] Karol Gregor，Andriy Mnih和Daan Wierstra。深度自回归网络。arXiv预印本arXiv：1310.8499，2013年。

[HBWP13] Matthew D Hoffman，David M Blei，Chong Wang和John Paisley。随机变分推论。机器学习研究杂志，14（1）：1303-1347，2013年。

[HDFN95] Geoffrey E Hinton，Peter Dayan，Brendan J Frey和Radford M Neal。无监督神经网络的“唤醒睡眠”算法。科学，第1158-1158页，1995年。

[KRL08] Koray Kavukcuoglu，Marc'Aurelio Ranzato和Yann LeCun。稀疏编码算法的快速推断及其在对象识别中的应用。技术报告CBLLTR-2008-12-01，纽约大学库兰特研究所计算和生物学习实验室，2008年。

[*Lin89]拉尔夫·林斯克。最大信息保存原理在线性系统中的应用。摩根·考夫曼出版社，1989年。*

[RGB13] Rajesh Ranganath，Sean Gerrish和David M Blei。黑匣子变异推理。

*arXiv预印本arXiv：1401.0118，2013年。*

[RMW14] Danilo Jimenez Rezende，Shakir Mohamed和Daan Wierstra。深潜高斯模型中的随机反向传播和变分推断。arXiv预印本arXiv：1401.4082，2014。

[Row98] Sam Roweis。用于PCA和SPCA的EM算法。神经信息处理系统的进展，第626-632页，1998年。

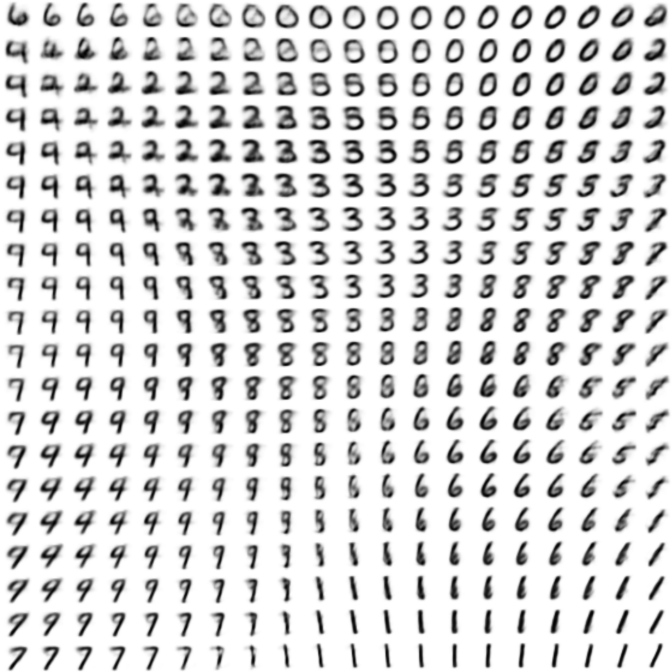
[SK13] Tim Salimans和David A Knowles。通过随机线性回归进行固定形式的后验逼近。贝叶斯分析，8（4），2013。

[SL10] Ruslan Salakhutdinov和Hugo Larochelle。高效学习深层Boltzmann机器。在国际人工智能与统计会议上，第693-700页，2010年。

[VLL+10] Pascal Vincent，Hugo Larochelle，Isabelle Lajoie，Yoshua Bengio和Pierre-Antoine Manzagol。堆叠式降噪自动编码器：在具有本地降噪标准的深度网络中学习有用的表示形式。机器学习研究杂志，9999：3371–3408，2010年。

# 可视化

看数字[4](#_bookmark30) 和[5](#_bookmark31) 可视化通过SGVB学习的模型的潜在空间和相应的观察空间。

* 1. 学到的Frey Face流形 （b）学到的MNIST流形

图4：通过AEVB学习的具有二维潜在空间的生成模型的学习数据流形的可视化。由于潜在空间的先验是高斯，因此通过高斯的逆CDF变换单位平方上的线性间隔坐标以产生潜变量z的值。对于这些z值，我们绘制了相应的生成

*带*有学习参数θ的pθ（x | z）。

（a）二维潜在空间 （b）5维潜在空间 （c）10维潜在空间 （d）20维潜在空间

图5：从MNIST生成模型中获得的不同样本潜在空间维数的随机样本。

1. **-D***l***（q*φ*（z）|| p*θ*（z））的解，高斯情况**

变异下限（要最大化的目标）包含一个KL项，该项通常可以进行分析集成。在此，当先验pθ（z）= N（0，I）和后验近似qφ（z | x(i)）均为高斯时，我们给出解决方案。令J为z的维数。令μ和σ

表示在数据点i处求出的变化均值和sd，而让µj和σj分别表示

*这*些向量的第j个元素。然后：

r r

*q*θ（z）对数p（z）dz =

*n*（z，γ，α2）log n（z；0，i）dz

*J* 1 *J*

\ 2 2

=- 2log（2π）- 2

*j*=1

(µ*j*+ σ*j*)

和：

r r

*q*θ（z）对数qθ（z）dz =

*n*（z，ω，*q* ）log n（z，γ，，*q* ）dz

*J* 1 *J*

*− −*

*j*

因此：

= 对数（2π） \

2 2

*j*=1

（1 +对数σ2）

r

*d*kl（（q）（z）πθ（z）＝

1

=

2

*q*θ（z）（对数pθ（z）-logqθ（z））dz

*J*

\ (1 + log（（σj）2）-（µj）2-（σj）2)

*j*=1

当使用识别模型qφ（z | x）时，μ和sdσ就是x的函数，而

如本文中所例示的，变分参数φ。

# MLP作为概率编码器和解码器

在变分自动编码器中，神经网络被用作概率编码器和解码器。编码器和解码器有许多可能的选择，具体取决于数据和模型的类型。在我们的示例中，我们使用了相对简单的神经网络，即多层感知器（MLP）。对于编码器，我们使用带有高斯输出的MLP，而对于解码器，我们使用带有高斯或伯努利输出的MLP，具体取决于数据类型。

## 伯努利MLP作为解码器

在这种情况下，令pθ（x | z）是一个多元伯努利，其概率是通过具有单个隐藏层的全连接神经网络根据z计算的：

*D*

log p（xz z）＝ \X-log yi+（1～Xi）·log（1～Ⅰ）

*i*=1

其中y =fσ（W2 tanh（W1z + b1）+ b2） （11）其中，fσ（。）是元素S型sigmoid激活函数，其中θ= {W1，W2，b1，b2}是MLP的权重和偏差。

## 高斯MLP作为编码器或解码器

在这种情况下，让编码器或解码器是具有对角协方差结构的多元高斯：

log p（xz z）＝log n（x；γ，α2i）

其中µ = W4h + b4

对数σ2= W5h + b5

**h** = tanh（W3z = b3） (12)

其中{W3，W4，W5，b3，b4，b5}是用作解码器时MLP和部分θ的权重和偏差。请注意，当此网络用作编码器qφ（z | x）时，z和x会交换，

权重和偏差是变化参数φ。

# 边际似然估计器

我们推导了以下边际似然估计器，只要采样空间的维数较低（小于5维），就可以对边际似然进行良好的估计，并且

抽取足够的样本。令pθ（x，z）=pθ（z）pθ（x | z）是我们要采样的生成模型

对于给定的数据点x(i)，我们想要估计边际似然pθ（x(i)）。

估算过程包括三个阶段：

* 1. 使用gradientz logpθ（z | x）=∇zlogpθ（z）+∇zlogpθ（x | z），使用基于梯度的MCMC（例如Hybrid Monte Carlo）从后部采样L值{z(l)} ）。
  2. 将密度估计器q（z）拟合到这些样本{z(l)}。
  3. 再次，从后验L个新值。插入这些样品以及已安装的

*q*（z），转化为以下估算器：

1 *L*

*p*（X”\

*q*(**z**(*l*))

\*−*1

哪里 **z**(l)〜pθ（z | x(i)）

估计量的推导：

*L*

*l*=1

*p*~（z）p（X）(i)Z(i)

1

*p*（X(i) 7））

(*q(z) dz*

=

*p*（X(i)）

( *p*（x (i)，z）

*p*（x(i)，z）

*q*(**z**) *d***z**

=

*p*（X）(i)

*q 2）p（xq 3），z）* *q*(**z**)

= *d***z**

*P*（X）*q*

r

= *p*（z）xr

*p（x*(i)*，z）*

*q*(**z**)

*d***z**

*p（x*(i)*，z）*

1 *L q*(**z**(*l*))

\

哪里 **z**(l)〜pθ（z | x(i)）

*"' L*

*l*=1

*p*~（z）p（X）(i)Z(l)

# 蒙特卡洛EM

蒙特卡罗EM算法不使用编码器，而是使用∇zlogpθ（z | x）=计算的后验梯度从潜变量的后验采样。

*∇*zlogpθ（z）+∇zlogpθ（x | z）。蒙特卡洛EM程序由10个HMC跳变组成

自动调整步长的步长，以使接受率为90％，然后是5

权重使用获取的样本更新步骤。对于所有算法，均使用Adagrad逐步调整参数（以及随附的退火进度表）来更新参数。

使用来自训练和测试集的前1000个数据点来估计边际可能性，对于每个数据点，使用带有4个跳跃步的Hybrid Monte Carlo方法，从潜在变量的后部采样50个值。

# 完整的VB

如本文所述，可以对参数θ和潜在变量z进行变分推断，这与我们在本文中所做的只是潜在变量不同。在这里，我们将得出这种情况的估计量。

令pα（θ）对于以上介绍的参数（由α设置）具有某些优先级。边际可能性可以写成：

log pα（x）＝dkl（qπ（θ）pα（θx））+L（ω；x） (13)

其中第一个RHS项表示与真实后验近似的KL散度，其中L（φ; X）表示边缘似然的变化下界：

r

*L*(***φ***; **X**) =

*q*ω（Th）（log p th\*（x）+ log pα（Th））log qω（Th）d(14)

注意，这是一个下限，因为KL散度是非负的。当近似值和真实后验精确匹配时，边界等于真实边际。项logpθ（X）由

各个数据点的边际似然的总和logpθ（X）= },N

*i*=1

对数pθ（x(i)），

每个都可以重写为：

log pθ（x(i)）＝dkl（qω（z x x(i)）p pθ（z x x(i)）+l（θ，ω；x(i)） (15)

再次，第一个RHS项是与真实后验近似的KL散度，并且

*L*（θ，φ; x）是数据点i的边缘似然的变化下界：

r

/ 1

*L*(***θ****,* ***φ***; **x**(*i*)) =

*q*（z（x））log p ^（x(i)z）+ log p thy（z）-log q（z x）dz(16)

对eqs RHS的期望[(14)](#_bookmark34) 和[(16)](#_bookmark34) 显然可以写成三个单独期望值的总和，其中第二个和第三个组成部分有时可以分析解决，例如

当pθ（x）和qφ（z | x）均为高斯时。出于一般性考虑，我们在这里假设

期望是棘手的。

在本节概述的某些温和条件下（见论文），为选定的近似后代

*q*φ（θ）和qφ（z | x）我们可以重新定义条件样本-〜qφ（z | x）为

**z**

**z** =gφ（E，x）其中 ***E*** *∼ p*(***E***) （17）在这里我们选择一个先验p（E）和一个函数gφ（E，x）使得以下成立：

-

r

/ 1

*L*(***θ****,* ***φ***; **x**(*i*)) =

r

=

*q*（z（x））log p ^（x(i)z）+ log p thy（z）-log q（z x）dz

*p*（e） /log p（x(i)z）+ log p thy（z）-log q（z）x*qω（z）*

**Z**= Gω（E，X*qω（z）* 9））

*E*(18)

对于近似后验qφ（θ）可以执行相同的操作：

***θ***-=hφ（ζ）与 ***ζ*** *∼ p*(***ζ***) (19)

与上面类似，在这里我们选择一个先验p（ζ）和一个函数hφ（ζ），使得以下条件成立：

r

*L*(***φ***; **X**) =

r

=

*q*ω（Th）（log p th\*（x）+ log pα（Th））log qω（Th）d

*P*（ZEI）（log p th\*（x）+log pα（Th））log qω（Th）

***Th＝Hω（ZEI）***

*D-泽尔*(20)

为了简明起见，我们引入了一种简写形式fφ（x，z，θ）：

*f*φ（x，z，θ）= N·（logpθ（x | z）+ logpθ（z）-logqφ（z | x））+ logpα（θ）-logqφ（θ） (21)

使用方程式[(20)](#_bookmark34) 和[(18),](#_bookmark34) 给定数据点x(i)，变化下界的蒙特卡洛估计为：

1 *L*

\

*L*(***φ***; **X**) *"' L*

*l*=1

*fω（x）*

(*l*)

*Gω（E）*

(*l*)

*,* **x**(*l*)

（*H）*

(*l*)

)) (22)

其中E(l)〜p（E）和ζ(l)〜p（ζ）。估计量仅取决于显然不受φ影响的p（E）和p（ζ）中的样本，因此可以区分φ。所得的随机梯度可以与随机优化方法（例如SGD或Adagrad）结合使用[[dhs10]。](#_bookmark16) 查看算法[1](#_bookmark0) 用于计算随机梯度的基本方法。

## F.1范例

令先验参数和潜变量为中心各向同性高斯pα（θ）=

*N*（z; 0，I）和pθ（z）= N（z; 0，I）。请注意在这种情况下，先验缺少参数。让我们也

假设真实后验近似为高斯，对角协方差约为

ance。在这种情况下，我们可以让变分近似后验是具有对角协方差结构的多元高斯：

对数qω（Th）＝log n（Th）；***θ***，2I

***θ***

log q（z）x＝log n（z；**z**，2i） (23)

**z**

**算**法2伪代码，用于使用我们的估算器来计算随机梯度。有关功能fφ，gφ和hφ的含义，请参见文本。

**要**求：φ（变化参数的当前值）

**g** *←* 0

**因为l是1到L**

**x**←来自数据集XE的随机抽取←来自先前p（E）ζ的随机抽取←来自先前p（ζ）的随机抽取

**G**~（g）+ 1λf fω（x，gω（e，x），hω（ZEZ））

*L*

## 结束于

**返回g**

其中，μ**z**和σz尚未定义为x的函数。由于它们是高斯​​分布，我们可以参数化变分近似后验：

*qφ（θ）为*

***T***h-＝γ***θ***＋Sigth~（8）ZE.哪里 ***ζ*** *∼ N* (**0***,* **I**)

*qφ（z | x）为* =**z**+西格玛Z 8 E哪里 ***E*** *∼ N* (**0***,* **I**)

**z**

用8表示元素乘积。这些可以插入上面定义的下限（等式[(21)](#_bookmark35) 和[(22)).](#_bookmark35)

在这种情况下，可以构建方差较低的替代估计量，因为在此模型中，pα（θ），pθ（z），qφ（θ）和qφ（z | x）是高斯分布，因此fφ的四项可以解决

分析地。结果估计量为：

 *J*

*L*

1 \ 1 \ /

(*l*) 2

(*l*) 2

(*l*) 21

(*i*)



(*i*)

*L*(***φ***; **X**) *"' L*

*l*=1

*N ·*  2

*j*=1

1＋log（（西格玛**z,j**））（**z,j**）

*−* (σ**z,j**)

+对数pθ（x

**z** )

+ 1 \

*J*

/1 + log（（σ(l)）2）-（µ(l)）2-（σ(l)）21

(24)

*µ*(*i*)

(*i*)

2

*j*=1

***θ****,j*

***θ****,j*

***θ****,j*

*j*和σ*j*仅表示向量μ(i)和σ(i)的第j个元素。