变分贝叶斯自动编码

Diederik P. Kingma
Machine Learning Group
Universiteit van Amsterdam
dpkingma@gmail.com

Max Welling
Machine Learning Group
Universiteit van Amsterdam
welling.max@gmail.com

摘 要

如果存在具有难治性后验分布的连续潜变量和大数据集，我们如何在定向概率模型中进行有效的推理和学习？ 我们引入了一种随机变分推理和学习的算法，可以扩展到大型数据集，并且在一些温和的差异条件下，甚至在棘手的情况下工作。 我们的贡献是双重的。首先，我们证明了变分下界的重新参数化产生了一个下限估计，可以使用标准随机梯度方法直接优化。 其次，我们展示了 i.i.d.（独立同分布的简写，下同） 且每个数据点具有连续潜变量的数据集，通过使用所提出的下界估计器将近似推理模型（也称为识别模型）拟合到难治性后验，可以使后验推断特别有效。 理论优势反映在实验结果中。

1 介绍

我们如何使用定向概率模型进行有效的近似推理和学习，当其具有连续潜变量和/或参数具有难治性的后验分布？变分贝叶斯（VB）方法涉及对难治性后验的近似优化。不幸的是，常见的平均场方法需要分析预期的解决方案 w.r.t. 近似的后验，在一般情况下也是难以处理的。我们展示了变分下界的重新参数化如何产生下界的简单可微分无偏估计，该 SGVB（随机梯度变分贝叶斯）估计器可用于几乎任何具有连续潜变量和/或参数难治的模型中的有效近似后验推断，并且使用标准随机梯度上升技术可以直接进行优化。

对于 i.i.d. 的情况。数据集和每个数据点的连续潜在变量，我们提出了变分贝叶斯自动编码（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进行推理和学习，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，从而使我们能够有效地学习模型参数，而无需每个数据点都需要昂贵的迭代推理方案（如 MCMC）。学习的近似后验推理模型还可以用于许多任务，例如识别，去噪，表示和可视化目的。当神经网络用于识别模型时，我们得到变分自动编码器。

2 方法

本节中的策略可用于导出具有连续潜变量的各种有向图模型的下界估计（随机目标函数）。 我们将在这里局限于我们有 i.i.d 的常见情况。 每个数据点具有潜在变量的数据集，以及我们希望对（全局）参数执行最大似然（ML）或最大后验（MAP）推断以及潜在变量的变分推断的数据集。例如，它是

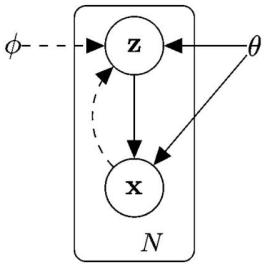


图 1：正在考虑的有向图模型的类型。 实线表示生成模型$p_{\boldsymbol{\theta}}(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$，虚线表示对难治性后验$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$的变分近似$q_{\phi}(\mathbf{z}|\mathbf{x})$。 变分参数 φ 与生成模型参数θ一起学习。

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; 该算法放在附录中，但该案例的实验留待将来工作。请注意，我们的方法可以应用于在线的，非静态的数据集，例如，流数据，但为了简单起见，我们假设一个固定的数据集。

2.1 问题场景

让我们考虑一些数据集$\mathbf{X} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$由 N i.i.d 组成，采样于一些连续或离散变量 x 的样本。我们假设数据是由一些随机过程生成的，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）从某些先验分布$p_{\boldsymbol{\theta}^*}(\mathbf{z})$生成值$\mathbf{z}^{(i)}$；（2）从某些条件分布$p_{\boldsymbol{\theta}^*}(\mathbf{x}|\mathbf{z})$生成值$\mathbf{x}^{(i)}$。 我们假设先验$p_{\boldsymbol{\theta}^*}(\mathbf{z})$和似然$p_{\boldsymbol{\theta}^*}(\mathbf{x}|\mathbf{z})$来自分布参数族$p_{\boldsymbol{\theta}}(\mathbf{z})$和$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$，并且它们的 PDF 几乎无处不在 w.r.t. θ 和 z。 不幸的是，我们的发现隐藏了很多这个过程：我们不知道真实参数$\boldsymbol{\theta}^*$以及潜在变量$\mathbf{z}^{(i)}$的值。

非常重要的是，我们不会对边际概率或后验概率做出常见的简化假设。 相反，我们在这里感兴趣的是一种通用算法，即使在以下情况下也能有效地工作：

1.难以处理：边际似然度积分$p_{\boldsymbol{\theta}}(\mathbf{x}) = \int p_{\boldsymbol{\theta}}(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\, d\mathbf{z}$的情况是难以处理的（所以我们不能评估或区分边际似然度），其中真实的后验密度$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})/p_{\boldsymbol{\theta}}(\mathbf{x})$是难以处理的（因此不能使用 EM 算法），并且任何合理的平均场 VB 算法所需的积分也是难以处理的。这些难以处理的情况非常普遍，并且出现在中等复杂似然函数$p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$的情况下，例如，具有非线性隐藏层的神经网络。

2.一个大型数据集：我们有如此多的数据，批量优化成本太高; 我们想使用小型微型计算机甚至单个数据点进行参数更新。 基于采样的解决方案，例如 蒙特卡罗 EM 一般来说太慢，因为它涉及每个数据点通常代价高昂的采样循环。

我们对上述场景中的三个相关问题感兴趣并提出解决方案:

1.参数θ的有效近似 ML 或 MAP 估计。 参数可以是它们自己感兴趣的，例如， 如果我们正在分析一些自然过程。它们还允许我们模仿隐随机过程并生成类似于真实数据的人工数据。

2.对于参数θ的选择给出观察值 x 的潜在变量 z 的有效近似后推断。 这对编码或数据表示任务很有用。

3.变量 x 的有效近似边际推断。 这允许我们执行各种推理任务，不过需要先验的 x。 计算机视觉中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，我们引入一个识别模型$q_\phi(\mathbf{z}|\mathbf{x})$：对难以处理的真后验$p_\theta(\mathbf{z}|\mathbf{x})$的近似。请注意，与平均场变分推断的近似后验相比，它不一定是阶乘的，并且其参数φ不是根据某些封闭形式的期望来计算的。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φ的方法。

从编码理论的角度来看，未观察到的变量 z 具有作为潜在表示或编码的价值。因此，在本文中，我们还将识别模型$q_\phi(\mathbf{z}|\mathbf{x})$称为概率编码器，因为给定数据点 x 它在编码 z 的可能值上产生分布（例如高斯分布），进而数据点 x 又可以从该分布中被生成出来。类似地，我们将$p_\theta(\mathbf{x}|\mathbf{z})$称为概率解码器，因为给定代码 z，它产生对 x 的可能对应值的分布。

2.2 变分界

边际似然度由各个数据点的边际似然度的总和组成$\log p_\theta(\mathbf{x}^{(1)},\cdots,\mathbf{x}^{(N)})=\sum_{i=1}^N \log p_\theta(\mathbf{x}^{(i)})$，可以被改写为：

$$\log p_\theta(\mathbf{x}^{(i)}) = D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)}) \tag{1}$$

第一个 RHS 术语是近似于真实后验的 KL 散度。 由于这个 KL-散度是非负的，所以第二个 RHS 项$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)})$被称为数据点 i 的边际似然（变分）下界，可以写成：

$$\log p_\theta(\mathbf{x}^{(i)}) \geq \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[-\log q_\phi(\mathbf{z}|\mathbf{x}) + \log p_\theta(\mathbf{x},\mathbf{z})\right] \tag{2}$$

也可以被重写为：

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)}) = -D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})\right] \tag{3}$$

我们想要区分和优化下界$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)})$ w.r.t. 变分参数 φ 和生成参数θ。 但是，下界 w.r.t. φ的梯度有点问题。 通常（na¨ıve）蒙特卡罗梯度估计这类问题用的方法是：

$\nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z})}\left[f(\mathbf{z})\right] = \mathbb{E}_{q_\phi(\mathbf{z})}\left[f(\mathbf{z})\nabla_{q_\phi(\mathbf{z})}\log q_\phi(\mathbf{z})\right] \simeq \frac{1}{L}\sum_{l=1}^L f(\mathbf{z})\nabla_{q_\phi(\mathbf{z}^{(l)})}\log q_\phi(\mathbf{z}^{(l)})$，其中$\mathbf{z}^{(l)} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$。

该梯度估计器表现出非常高的方差（参见例如[BJP12]）并且对于我们的目的是不实际的。

2.3 SGVB 估计器和 AEVB 算法

在本节中，我们将介绍下界及其衍生物 w.r.t 参数的实用估计。我们假设$q_\phi(\mathbf{z}|\mathbf{x})$形式的近似后验，但是请注意该技术可以应用于$q_\phi(\mathbf{z})$的情况，即我们不考虑条件 x。 附录中给出了用于推断参数后验的完全变分贝叶斯方法。

在 2.4 节中针对选定的近似后验$q_\phi(\mathbf{z}|\mathbf{x})$概述的某些温和条件下，我们可以使用（辅助）噪声变量ϵ的可微变换$g_\phi(\boldsymbol{\epsilon},\mathbf{x})$重新参数化随机变量$\widetilde{\mathbf{z}} \sim q_\phi(\mathbf{z}|\mathbf{x})$：

$$\widetilde{\mathbf{z}} = g_\phi(\boldsymbol{\epsilon},\mathbf{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) \tag{4}$$

有关选择此类适当分布$p(\boldsymbol{\epsilon})$和函数$g_\phi(\boldsymbol{\epsilon},\mathbf{x})$的一般策略，请参阅第 2.4 节。我们现在可以形成蒙特卡罗对某些函数$f(\mathbf{z})$ w.r.t $q_\phi(\mathbf{z}|\mathbf{x})$的期望估计如下：

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})}\left[f(\mathbf{z})\right] = \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[f(g_\phi(\boldsymbol{\epsilon},\mathbf{x}^{(i)}))\right] \simeq \frac{1}{L}\sum_{l=1}^L f(g_\phi(\boldsymbol{\epsilon}^{(l)},\mathbf{x}^{(i)})) \quad \text{where} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon}) \tag{5}$$

我们将这种技术应用于变分下界（方程（2）），得到我们的通用随机梯度变分贝叶斯（SGVB）估计量$\widetilde{\mathcal{L}}^A(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)}) \simeq \mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)})$：

$$\begin{aligned}\widetilde{\mathcal{L}}^A(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)}) &= \frac{1}{L}\sum_{l=1}^L \log p_\theta(\mathbf{x}^{(i)},\mathbf{z}^{(i,l)}) - \log q_\phi(\mathbf{z}^{(i,l)}|\mathbf{x}^{(i)}) \\ \text{where} \quad \mathbf{z}^{(i,l)} &= g_\phi(\boldsymbol{\epsilon}^{(i,l)},\mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})\end{aligned} \tag{6}$$

算法 1 变分贝叶斯自动编码（AEVB）算法的 Minibatch 版本。 可以使用 2.3 节中的两个 SGVB 估计中的任何一个。 我们在实验中使用设置 M = 100 和 L = 1。

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ 初始化参数

重复

$\mathbf{X}^M \leftarrow$ M 数据点的随机小批量（从完整数据集中提取）

$\boldsymbol{\epsilon} \leftarrow$ 来自噪声分布的随机样本 p（$\boldsymbol{\epsilon}$）

$\mathbf{g} \leftarrow \nabla_{\boldsymbol{\theta},\boldsymbol{\phi}} \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M, \boldsymbol{\epsilon})$ (小批量估算器的梯度 (8))

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ 使用梯度 g 更新参数（例如 SGD 或 Adagrad [DHS10]）

直到 参数收敛 ($\boldsymbol{\theta}, \boldsymbol{\phi}$)

返回 $\boldsymbol{\theta}, \boldsymbol{\phi}$;

通常，方程的 KL-散度 $D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}))$ （3）可以通过分析进行积分（见附录 B），这样只有预期的重建误差 $\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})\right]$ 才需要通过采样进行估计。 然后可以将 KL-散度解释为正则化 φ，从而鼓励近似后验接近先前的 $p_{\boldsymbol{\theta}}(\mathbf{z})$。 这产生 SGVB 估计器 $\widetilde{\mathcal{L}}^B(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \simeq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$ 的第二版本，对应于等式（3），其通常具有比通用估计器更小的方差：

$$\widetilde{\mathcal{L}}^B(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \frac{1}{L}\sum_{l=1}^{L}(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}))$$
$$\text{where} \quad \mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon}) \tag{7}$$

给定来自具有 N 个数据点的数据集 X 的多个数据点，我们可以基于小批量构建完整数据集的边际似然下界的估计：

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}) \simeq \widetilde{\mathcal{L}}^M(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{X}^M) = \frac{N}{M}\sum_{i=1}^{M}\widetilde{\mathcal{L}}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \tag{8}$$

其中小批量 $\mathbf{X}^M = \{\mathbf{x}^{(i)}\}_{i=1}^M$ 是来自具有 N 个数据点的完整数据集 X 的 M 个数据点的随机抽取样本。在我们的实验中，我们发现每个数据点的样本数 L 可以设置为 1，只要小批量大小 M 足够大，例如， M = 100。衍生物 $\nabla_{\boldsymbol{\theta},\boldsymbol{\phi}}\widetilde{\mathcal{L}}(\boldsymbol{\theta}; \mathbf{X}^M)$ 可以采用，并且得到的梯度可以与诸如 SGD 或 Adagrad [DHS10]的随机优化方法结合使用。有关计算随机梯度的基本方法，请参阅算法 1。

当查看方程式给出的目标函数时，与自动编码器的连接变得清晰（7）。第一项是（先验的近似后向的 KL 散度）充当正则化器，而第二项是预期的负重建误差。选择函数 $g_{\boldsymbol{\phi}}(.)$，使得它将数据点 $\mathbf{x}^{(i)}$ 和随机噪声向量 $\boldsymbol{\epsilon}^{(l)}$ 映射到该数据点的近似后验的样本：$\mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)}, \mathbf{x}^{(i)})$ 其中 $\mathbf{z}^{(i,l)} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})$。随后，将样本 $\mathbf{z}^{(i,l)}$ 输入到函数 $\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$，其等于生成模型下数据点 $\mathbf{x}^{(i)}$ 的概率密度（或质量），在给定 $\mathbf{z}^{(i,l)}$ 时。该术语是自动编码器用语中的负重建误差。

2.4 重新参数化技巧

为了解决我们的问题，我们调用了另一种从 $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ 生成样本的方法。 基本的参数化技巧非常简单。 设 z 是连续随机变量，$\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ 是一些条件分布。 然后经常可以将随机变量 z 表示为确定性变量 $\mathbf{z} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x})$，其中 ϵ 是具有独立边际 $p(\boldsymbol{\epsilon})$ 的辅助变量，并且 $g_{\boldsymbol{\phi}}(.)$ 是φ参数化的一些向量值函数。

这种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 w.r.t $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$，这样蒙特卡罗估计的期望是可微分的 w.r.t. φ。证明如下。 鉴于确定性映射 $\mathbf{z} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x})$，从而我们知道 $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\prod_i dz_i = p(\boldsymbol{\epsilon})\prod_i d\epsilon_i$。 因此（见页底 1），$\int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) f(\mathbf{z})\, d\mathbf{z} = \int p(\boldsymbol{\epsilon}) f(\mathbf{z})\, d\boldsymbol{\epsilon} = \int p(\boldsymbol{\epsilon}) f(g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}))\, d\boldsymbol{\epsilon}$。

[1] 请注意，对于无穷小，我们使用符号约定 $d\mathbf{z} = \prod_i dz_i$

随后可以构造可微分估计：$\int q_\phi(\mathbf{z}|\mathbf{x})f(\mathbf{z})\,d\mathbf{z} \simeq \frac{1}{L}\sum_{l=1}^{L} f(g_\phi(\mathbf{x},\boldsymbol{\epsilon}^{(l)}))$，其中$\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$。 在 2.3 节中，我们应用这个技巧来获得变分下界的可微分估计。

例如，采用单变量高斯情形：设$z \sim p(z|x) = \mathcal{N}(\mu, \sigma^2)$。 在这种情况下，有效的重新参数化是$z = \mu + \sigma\epsilon$，其中ϵ是辅助噪声变量$\epsilon \sim \mathcal{N}(0,1)$。 因此，

我们可以如何为$q_\phi(\mathbf{z}|\mathbf{x})$选择这样一个可微分变换$g_\phi(.)$和辅助变量$\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$？ 三种基本方法是:

1. 可追踪逆 CDF。 在这种情况下，设$\boldsymbol{\epsilon} \sim \mathcal{U}(\mathbf{0},\mathbf{I})$，并且令$g_\phi(\boldsymbol{\epsilon},\mathbf{x})$为$q_\phi(\mathbf{z}|\mathbf{x})$的逆 CDF。 示例： Exponential，Cauchy，Logistic，Rayleigh，Pareto，Weibull，Reciprocal，Gompertz，Gumbel 和 Erlang 分布。
2. 类似于高斯示例，对于任何“定位（location） - 规模（scale）”分布族，我们可以选择标准分布（定位= 0，规模= 1）作为辅助变量ϵ，并且让$g(.) = \text{location} + \text{scale} \cdot \boldsymbol{\epsilon}$。 示例: Laplace, Elliptical，Student’s t，Logistic，Uniform，Triangular 和 Gaussian 分布。
3. 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：Log-Normal（正态分布变量的取幂），Gamma（指数分布变量的和），Dirichlet（Gamma 变量的加权和），Beta，Chi-Squared 和 F 分布。

当所有三种方法都失败时，存在对逆 CDF 的良好近似，要求计算具有与 PDF 相当的时间复杂度（对于某些方法，参见例如[Dev86]）。

3 示例：变分自动编码器

在本节中，我们将给出一个示例，其中我们使用神经网络作为概率编码器$q_\phi(\mathbf{z}|\mathbf{x})$（对生成模型$p_\theta(\mathbf{x},\mathbf{z})$的后验的近似）以及使用 AEVB 算法对参数φ和θ联合优化。

让先验的潜在变量为中心各向同性多变量高斯$p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z};\mathbf{0},\mathbf{I})$。请注意，在这种情况下，先验缺少参数。我们让$p_\theta(\mathbf{x}|\mathbf{z})$是多元高斯（在实值数据的情况下）或伯努利（在二进制数据的情况下），其分布参数是从 z 计算的 MLP（一个完全连接的神经网络与单个隐藏层，见附录 C）。注意真正的后验$p_\theta(\mathbf{z}|\mathbf{x})$在这种情况下是难以处理的。虽然$q_\phi(\mathbf{z}|\mathbf{x})$形式有很多自由度，但我们假设真实的（但难以处理的）后验采用具有近似对角线协方差的近似高斯形式。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(见页底 2):

$$\log q_\phi(\mathbf{z}|\mathbf{x}^{(i)}) = \log \mathcal{N}(\mathbf{z};\boldsymbol{\mu}^{(i)},\boldsymbol{\sigma}^{2(i)}\mathbf{I}) \tag{9}$$

其中的平均值和 s.d. 近似后验，$\boldsymbol{\mu}^{(i)}$和$\boldsymbol{\sigma}^{(i)}$是编码 MLP 的输出，即数据点$\mathbf{x}^{(i)}$的非线性函数和变分参数φ（见附录 C）。

如 2.4 节所述，我们使用$\mathbf{z}^{(i,l)} = g_\phi(\mathbf{x}^{(i)},\boldsymbol{\epsilon}^{(l)}) = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)}$从后验$\mathbf{z}^{(i,l)} \sim q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$进行采样，其中$\boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(\mathbf{0},\mathbf{I})$，$\odot$表示按元素乘积。 在该模型中，$p_\theta(\mathbf{z})$（先验）和$q_\phi(\mathbf{z}|\mathbf{x})$都是高斯分布; 在这种情况下，我们可以使用方程（7）的估计量。可以在不估计的情况下计算和区分 KL 散度（见附录 B）。 此模型和数据点$\mathbf{x}^{(i)}$的结果估计值为:

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi};\mathbf{x}^{(i)}) \simeq \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\right) + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$$
$$\text{where} \quad \mathbf{z}^{(i,l)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\sigma}^{(i)} \odot \boldsymbol{\epsilon}^{(l)} \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim \mathcal{N}(0,\mathbf{I}) \tag{10}$$

如上所述和附录 C 中所述，解码项$\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$是伯努利还是高斯 MLP，具体取决于我们建模的数据类型。

[2] 请注意，这只是一个（简化）选择，而不是我们方法的限制。

4 相关工作

据我们所知，唤醒睡眠算法[HDFN95]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。与我们的方法一样，唤醒睡眠算法采用近似真实后验的识别模型。唤醒 - 睡眠算法的缺点是它需要同时优化两个目标函数，这两个目标函数一起不对应于边际似然（边界）的优化。唤醒睡眠的一个优点是它也适用于具有离散潜变量的模型。唤醒 - 睡眠与每个数据点的 AEVB 具有相同的计算复杂度。

随机变分推断[HBWP13]最近受到越来越多的关注。最近，[BJP12]引入了一个控制变量方案，以减少 2.1 节中讨论的天线梯度估计的高方差，并应用于后期的指数族近似。在[RGB13]中，引入了一些通用方法，即控制变量方案，用于减小原始梯度估计器的方差。在[SK13]中，类似于本文的重新参数化被用于随机变分推理算法的有效版本中，用于学习指数族近似分布的自然参数。

AEVB 算法暴露了定向概率模型（用变分物镜训练）和自动编码器之间的连接。线性自动编码器和某类生成线性高斯模型之间的联系早已为人所知。在[Row98]中，显示 PCA 对应于具有先验 $p(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$和条件分布$p(\mathbf{x}|\mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z}, \epsilon\mathbf{I})$的线性高斯模型的特殊情况的最大似然（ML）解，特别是无穷小ϵ的情况。

在最近关于自动编码器[VLL + 10]的相关工作中，显示了非正则化自动编码器的训练标准对应于输入 X 和潜在的互信息的下界的最大化（参见 infomax 原理[Lin89]）。表示互信息的最大化（wrt 参数）相当于最大化条件反射，条件反射受自动编码模型[VLL + 10]数据的预期对数似然的限制，即负重建误差。然而，众所周知，这种重建标准本身不足以学习有用的表示[BCV13]。已经提出了规则化技术来使自动编码器学习有用的表示，例如去噪，收缩和稀疏自编码变体[BCV13]。 SGVB 目标包含由变分界限决定的正则化项（例如方程（10）），缺乏学习有效表示所需的通常的噪音调节超参数。相关的还有编码器 - 解码器架构，例如预测稀疏分解（PSD）[KRL08]，我们从中得到了一些启发。最近引入的生成随机网络[BTL13]也是相关的，其中有噪声的自动编码器学习从数据分布中采样的马尔可夫链的转移算子。在[SL10]中，使用识别模型进行深度玻尔兹曼机器的有效学习。与我们提出的用于学习一般定向概率模型的算法相比，这些方法针对非标准化模型（即，诸如玻尔兹曼机器的无向模型）或限于稀疏编码模型。

最近提出的 DARN 方法[GMW13]也使用自动编码结构学习定向概率模型，但是它们的方法适用于二进制潜在变量。最近，[RMW14]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分推理之间建立了联系。他们的工作是独立于我们开发的，并为 AEVB 提供了额外的视角。

5 实验

我们训练了来自 MNIST 和 Frey Face 数据集（见页底 3）的图像的生成模型，并根据变分下界和估计边际似然度比较学习算法。

使用来自部分 3 的生成模型（编码器）和变分近似（解码器），其中所描述的编码器和解码器具有相等数量的隐藏单元。 由于 Frey Face 数据是连续的，我们使用了具有高斯输出的解码器，与编码器相同，除了在解码器输出使用 S 形激活函数将均值约束到区间（0，1）。

[3] 在 http://www.cs.nyu.edu/~roweis/data.html 可访问到

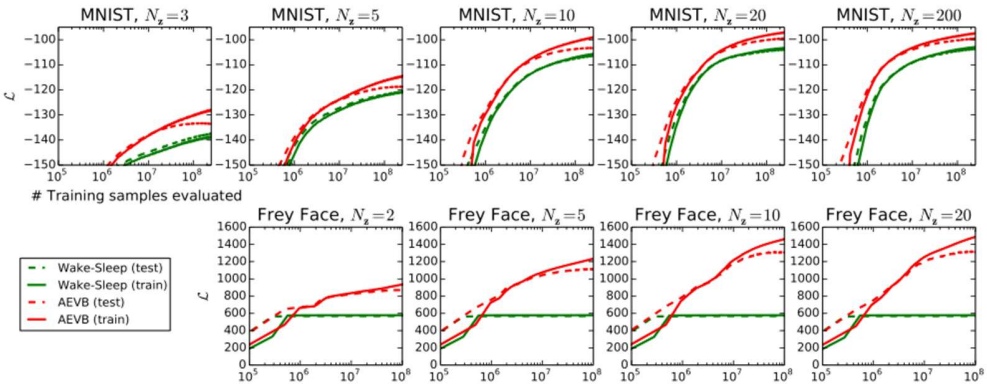


图 2：我们的 AEVB 方法与唤醒 - 睡眠算法的比较，在优化下限方面，针对潜在空间（N_z）的不同维度。 我们的方法收敛得更快，并在所有实验中达到了更好的解决效果。 有趣的是，更多的潜变量不会导致更多的过拟合，这可以通过下限的正则化效应来解释。 垂直轴：每个数据点的估计平均变分下限。 估计量方差很小（<1）并省略。 水平轴：评估的训练点数量。 计算每百万个训练样本大约需要 20-40 分钟，英特尔 Xeon CPU 的运行效率为 40 GFLOPS。

请注意，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器$\nabla_{\theta,\phi}\mathcal{L}(\theta,\phi;\mathbf{X})$来计算梯度（参见算法 1），加上对应于先前$p(\theta)=\mathcal{N}(0,\mathbf{I})$的小权重衰减项。该目标的优化等效于近似 MAP 估计，其中似然梯度由下界的梯度近似。

我们比较了 AEVB 与唤醒睡眠算法[HDFN95]的性能。我们为唤醒睡眠算法和变分自动编码器采用了相同的编码器（也称为识别模型）。通过从$\mathcal{N}(0,0.01)$随机抽样初始化所有变量和生成参数，并使用 MAP 标准联合随机优化。使用 Adagrad [DHS10]调整步骤;根据前几次迭代中训练集的性能，从{ 0.01，0.02，0.1 }中选择 Adagrad 全局步长参数。使用尺寸为 M = 100 的小批量，每个数据点 L = 1 个样品。

似然度下界。我们训练了生成模型（解码器）和相应的编码器（也就是识别模型），在 MNIST 的情况下有 500 个隐藏单元，在 Frey Face 数据集的情况下有 200 个隐藏单元（以防止过拟合，因为它是一个相当小的数据集）。所选择的隐藏单元数量基于先前关于自动编码器的文献，并且不同算法的相对性能对这些选择不是非常敏感。图 2 显示了比较下界时的结果。有趣的是，多余的潜变量并没有导致过拟合，这可以通过变分界的正则化性质来解释。

边际似然度。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似然度。有关边际似然估计的更多信息，请参见附录。对于编码器和解码器，我们再次使用神经网络，这次有 100 个隐藏单元和 3 个潜变量;对于更高维度的潜在空间，估计变得不可靠。同样，使用了 MNIST 数据集。将 AEVB 和 Wake-Sleep 方法与 Monte Carlo EM（MCEM）与混合蒙特卡罗（HMC）[DKPR87]采样器进行比较;详情见附录。我们比较了三种算法的收敛速度，适用于小型和大型训练集大小。结果如图 3 所示。

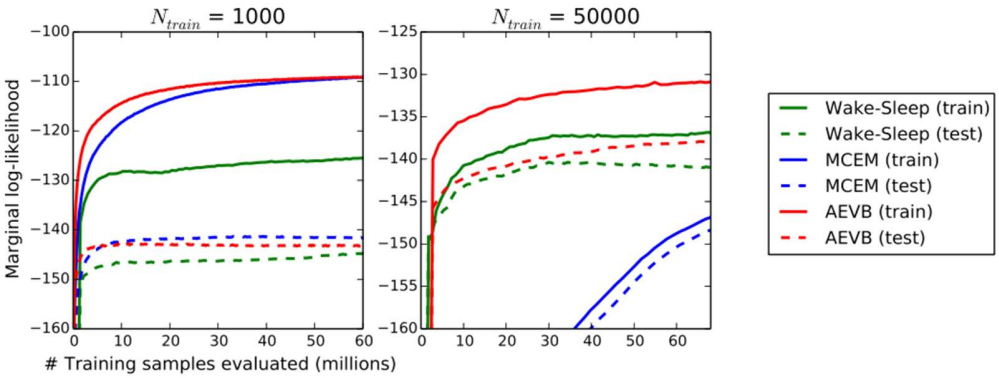


图 3：针对不同数量的训练点，就估计的边际似然度而言，AEVB 与唤醒 - 睡眠算法和蒙特卡罗 EM 的比较。 蒙特卡罗 EM 不是在线算法，并且（与 AEVB 和唤醒 - 睡眠方法不同）不能有效地应用于完整的 MNIST 数据集。

高维数据的可视化。 如果我们选择低维潜在空间（例如 2D），我们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。 有关 MNIST 和 Frey Face 数据集的 2D 潜在流形的可视化，请参见附录 A。

6 结论

我们引入了一个新的变分下界估计，变分贝叶斯随机梯度（SGVB），用于连续潜变量的有效近似推断。 可以使用标准随机梯度方法直接区分和优化所提出的估计器。 对于 i.i.d 的情况。 数据集和每个数据点的连续潜变量我们引入了一种有效推理和学习的算法，变分贝叶斯自动编码（AEVB），它使用 SGVB 估计器学习近似推理模型。 理论上的优势体现在实验结果中。

7 未来的工作

由于 SGVB 估计器和 AEVB 算法可以应用于几乎任何具有连续潜变量的推理和学习问题，因此有很多未来方向：（i）学习用于编码器的深度神经网络（例如卷积网络）的分层生成架构和解码器，与 AEVB 联合训练；（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）具有潜变量的监督模型，可用于学习复杂的噪声分布。

参考文献

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. 2013.

[BJP12] David M Blei, Michael I Jordan, and John W Paisley. Variational bayesian inference with stochastic search. In Proceedings of the 29th International Conference on Machine Learning (ICML-12), pages 1367–1374, 2012.

[BTL13] Yoshua Bengio and Eric Thibodeau-Laufer. Deep generative stochastic networks trainable by backprop. arXiv preprint arXiv:1306.1091, 2013.

[Dev86] Luc Devroye. Sample-based non-uniform random variate generation. In Proceedings of the 18th conference on Winter simulation, pages 260–265. ACM, 1986.

[DHS10] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 12:2121–2159, 2010.

[DKPR87] Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. Physics letters B, 195(2):216–222, 1987.

[GMW13] Karol Gregor, Andriy Mnih, and Daan Wierstra. Deep autoregressive networks. arXiv preprint arXiv:1310.8499, 2013.

[HBWP13] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. The Journal of Machine Learning Research, 14(1):1303–1347, 2013.

[HDFN95] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The" wake-sleep" algorithm for unsupervised neural networks. SCIENCE, pages 1158–1158, 1995.

[KRL08] Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. Fast inference in sparse coding algorithms with applications to object recognition. Technical Report CBLL-TR-2008-12-01, Computational and Biological Learning Lab, Courant Institute, NYU, 2008.

[Lin89] Ralph Linsker. An application of the principle of maximum information preservation to linear systems. Morgan Kaufmann Publishers Inc., 1989.

[RGB13] Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. arXiv preprint arXiv:1401.0118, 2013.

[RMW14] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic back-propagation and variational inference in deep latent gaussian models. arXiv preprint arXiv:1401.4082, 2014.

[Row98] Sam Roweis. EM algorithms for PCA and SPCA. Advances in neural information processing systems, pages 626–632, 1998.

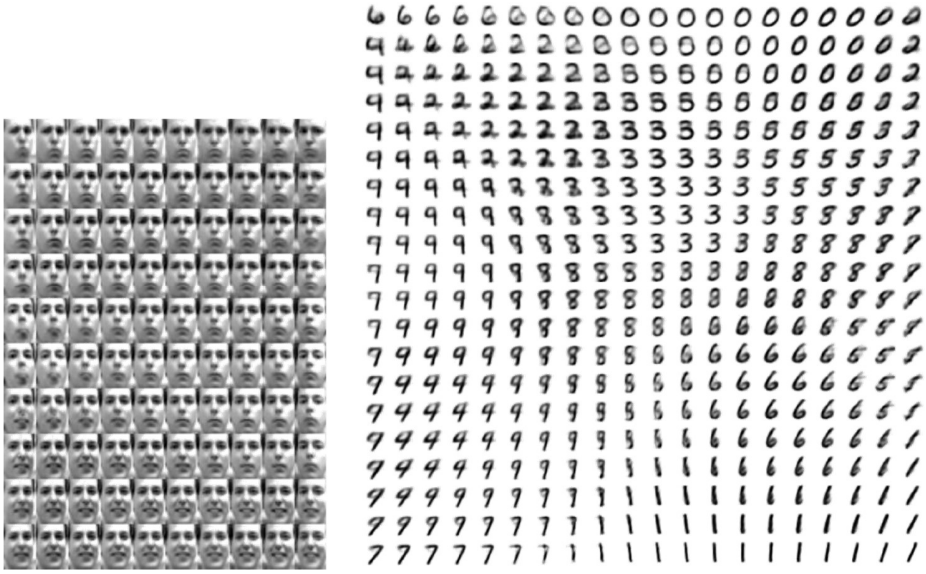
[SK13] Tim Salimans and David A Knowles. Fixed-form variational posterior approximation through stochastic linear regression. Bayesian Analysis, 8(4), 2013.

[SL10] Ruslan Salakhutdinov and Hugo Larochelle. Efficient learning of deep boltzmann machines. In International Conference on Artificial Intelligence and Statistics, pages 693–700, 2010.

[VLL+10] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. The Journal of Machine Learning Research, 9999:3371–3408, 2010.

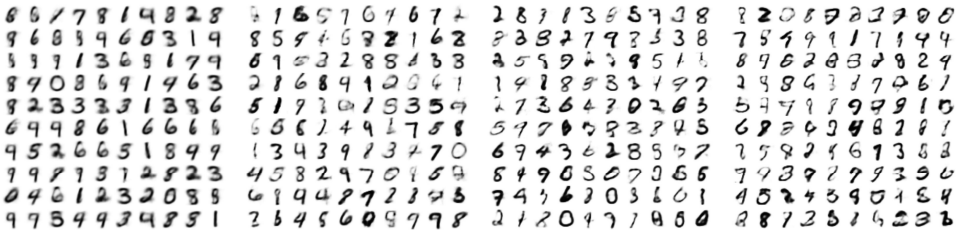
A 可视化

有关潜在空间的可视化以及使用 SGVB 学习的模型的相应观察空间，请参见图 4 和图 5。



(a) 学到的 Frey Face 流形 (b) 学到的 MNIST 流形

图 4：使用 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的，因此通过高斯的逆 CDF 变换单位平方上的线性间隔的坐标，以产生潜在变量 z 的值。 对于这些值 z 中的每一个，我们用学习的参数θ绘制相应的生成$p_\theta(\mathbf{x}|\mathbf{z})$。



(a) 2-D 潜在空间 (b) 5-D 潜在空间 (c) 10-D 潜在空间 (d) 20-D 潜在空间

图 5：来自 MNIST 的学习生成模型的随机样本，用于潜在空间的不同维度。

B $-D_{KL}(q_\phi(\mathbf{z})||p_\theta(\mathbf{z}))$的解，高斯情形

变分下界（要最大化的目标）包含一个 KL 术语，通常可以通过分析进行整合。 这里我们给出解决方案，当先验$p_\theta(\mathbf{z}) = \mathcal{N}(0, \mathbf{I})$和后验近似$q_\phi(\mathbf{z}|\mathbf{x}^{(i)})$都是高斯的。 设 J 为 z 的维数。 并且设μ和σ表示变分均值以及 s.d. 在数据点 i 处的评估，并且让μ_j和σ_j简单地表示这些向量的第 j 个元素。于是：

$$\int q_{\boldsymbol{\theta}}(\mathbf{z}) \log p(\mathbf{z})\, d\mathbf{z} = \int \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \log \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})\, d\mathbf{z}$$

$$= -\frac{J}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{J}(\mu_j^2 + \sigma_j^2)$$

并且:

$$\int q_{\boldsymbol{\theta}}(\mathbf{z}) \log q_{\boldsymbol{\theta}}(\mathbf{z})\, d\mathbf{z} = \int \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2) \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2)\, d\mathbf{z}$$
$$= -\frac{J}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{J}(1 + \log \sigma_j^2)$$

因此:

$$-D_{KL}((q_{\boldsymbol{\phi}}(\mathbf{z})||p_{\boldsymbol{\theta}}(\mathbf{z})) = \int q_{\boldsymbol{\theta}}(\mathbf{z})\left(\log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\theta}}(\mathbf{z})\right)\, d\mathbf{z}$$
$$= \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2\right)$$

当使用识别模型 $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ ，并且μ和 s.d. σ仅仅是 x 和变分参数φ的函数，如文中所示。

C MLP's 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。 编码器和解码器有很多种可能的选择，具体取决于数据和型号的类型。 在我们的例子中，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，我们使用具有高斯输出的 MLP，而对于解码器，我们使用具有高斯或伯努利输出的 MLP，这取决于数据的类型。

C.1 Bernoulli MLP 作为解码器

在这种情况下，让 $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ 为多变量伯努利，其概率是从 z 计算的，具有完全连接的神经网络和单个隐藏层:

$$\log p(\mathbf{x}|\mathbf{z}) = \sum_{i=1}^{D} x_i \log y_i + (1 - x_i) \cdot \log(1 - y_i)$$
$$\text{where } \mathbf{y} = f_\sigma(\mathbf{W}_2 \tanh(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2) \tag{11}$$

其中 $f_\sigma(.)$ 是元素范围（elementwise）的 sigmoid 激活函数，并且其中 $\boldsymbol{\theta} = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2\}$ 是 MLP 的权重和偏差。

C.2 Gaussian MLP 作为编码器或解码器

在这种情况下，让编码器或解码器成为具有对角协方差结构的多元高斯:

$$\log p(\mathbf{x}|\mathbf{z}) = \log \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\sigma}^2\mathbf{I})$$
$$\text{where } \boldsymbol{\mu} = \mathbf{W}_4\mathbf{h} + \mathbf{b}_4$$
$$\log \boldsymbol{\sigma}^2 = \mathbf{W}_5\mathbf{h} + \mathbf{b}_5$$
$$\mathbf{h} = \tanh(\mathbf{W}_3\mathbf{z} + \mathbf{b}_3) \tag{12}$$

其中当用作解码器的时候，$\{\mathbf{W}_3, \mathbf{W}_4, \mathbf{W}_5, \mathbf{b}_3, \mathbf{b}_4, \mathbf{b}_5\}$ 是 MLP 和部分θ的权重和偏差。 注意，当该网络用作编码器 $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ 时，则交换 z 和 x，并且权重和偏差是变分参数φ。

D 边际似然估计

我们推导出以下边际似然估计，只要采样空间的维数较低（小于 5 维）并且采集了足够的样本，就可以产生良好的边际似然估计。 设 $p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ 是我们从中采样的生成模型，对于给定的数据点 $\mathbf{x}^{(i)}$，我们想估计边际似然 $p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$。

估算过程包括三个阶段:

1. 使用基于梯度的 MCMC，例如来自后验的样本 L 值 $\{\mathbf{z}^{(l)}\}$，例如， 混合蒙特卡罗，使用 $\nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{z}) + \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$。
2. 将密度估计量 $q(\mathbf{z})$ 拟合到这些样本 $\{\mathbf{z}^{(l)}\}$。
3. 再次地，从后验采样 L 个新值。 将这些样本以及拟合的 $q(\mathbf{z})$ 插入以下估算器：

$$p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) \simeq \left(\frac{1}{L} \sum_{l=1}^{L} \frac{q(\mathbf{z}^{(l)})}{p_{\boldsymbol{\theta}}(\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(l)})} \right)^{-1} \quad \text{where} \quad \mathbf{z}^{(l)} \sim p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})$$

估算器的推导:

$$\begin{aligned}
\frac{1}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})} &= \frac{\int q(\mathbf{z})\, d\mathbf{z}}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})} = \frac{\int q(\mathbf{z}) \frac{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z})}\, d\mathbf{z}}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})} \\
&= \int \frac{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})} \frac{q(\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z})}\, d\mathbf{z} \\
&= \int p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)}) \frac{q(\mathbf{z})}{p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z})}\, d\mathbf{z} \\
&\simeq \frac{1}{L} \sum_{l=1}^{L} \frac{q(\mathbf{z}^{(l)})}{p_{\boldsymbol{\theta}}(\mathbf{z}) p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(l)})} \quad \text{where} \quad \mathbf{z}^{(l)} \sim p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})
\end{aligned}$$

E 蒙特卡罗 EM

蒙特卡罗 EM 算法不使用编码器，而是使用 $\nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{z}) + \nabla_{\mathbf{z}} \log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ 计算的后验梯度从潜在变量的后端进行采样。 蒙特卡罗 EM 程序包括 10 个 HMC 跳跃步骤，其自动调整步长，使得接受率为 90%，然后使用所获取的样本进行 5 个权重更新步骤。 对于所有算法，使用 Adagrad 步长（伴随退火计划）更新参数。

使用来自训练和测试集的前 1000 个数据点估计边际似然度，对于每个数据点，使用具有 4 个“蛙跳”步骤的混合蒙特卡罗从潜在变量的后部采样 50 个值。

F 完全变分贝叶斯

如本文所述，可以对参数θ和潜在变量 z 进行变分推理，而不像我们在论文中所做的那样只是潜在变量。 在这里，我们将为该情形推导出我们的估算器。

设 $p_{\boldsymbol{\alpha}}(\boldsymbol{\theta})$ 为上面介绍的参数的一些高优先级，参数化为α。 边际似然度可写为：

$$\log p_{\boldsymbol{\alpha}}(\mathbf{X}) = D_{KL}(q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) || p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}|\mathbf{X})) + \mathcal{L}(\boldsymbol{\phi}; \mathbf{X}) \tag{13}$$

其中第一个 RHS 项表示近似值与真实后验的 KL 偏差，其中 $\mathcal{L}(\boldsymbol{\phi}; \mathbf{X})$ 表示边际似然的变分下界：

$$\mathcal{L}(\boldsymbol{\phi}; \mathbf{X}) = \int q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \left(\log p_{\boldsymbol{\theta}}(\mathbf{X}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \right) d\boldsymbol{\theta} \tag{14}$$

请注意，这是一个下限，因为 KL 散度是非负的; 当近似和真实的后验完全匹配时，界限等于真正的边际。 术语 $\log p_{\boldsymbol{\theta}}(\mathbf{X})$ 由各个数据点 $\log p_{\boldsymbol{\theta}}(\mathbf{X}) = \sum_{i=1}^{N} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})$ 的边际似然度的总和组成，每个数据点可以重写为：

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)}) || p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) \tag{15}$$

其中第一个 RHS 术语是近似于真实后验的 KL 散度，而$\mathcal{L}(\boldsymbol{\theta},\phi;\mathbf{x})$是数据点 i 的边际似然的变分下界：

$$\mathcal{L}(\boldsymbol{\theta},\phi;\mathbf{x}^{(i)}) = \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\left(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right) d\mathbf{z} \tag{16}$$

对方程（14）和（16）的 RHS 的期望显然可以写成三个单独期望的总和，其中第二和第三分量有时可以被分析出来从而解决，例如， 当$p_{\boldsymbol{\theta}}(\mathbf{x})$和 $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$都是高斯分布时。 为了一般性，我们在此假设这些期望中的每一个都是难以处理的。

在选定的近似后验$q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$和$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$的部分（见论文）中概述的某些温和条件下，我们可以将条件样本$\widetilde{\mathbf{z}} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$重新参数化为

$$\widetilde{\mathbf{z}} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\mathbf{x}) \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) \tag{17}$$

我们选择先验的$p(\boldsymbol{\epsilon})$和函数$g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\mathbf{x})$，以便满足以下条件：

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta},\phi;\mathbf{x}^{(i)}) &= \int q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\left(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right) d\mathbf{z} \\ &= \int p(\boldsymbol{\epsilon})\left(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\right)\Big|_{\mathbf{z}=g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon},\mathbf{x}^{(i)})} d\boldsymbol{\epsilon}\end{aligned} \tag{18}$$

对于近似后验$q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$也可以这样做：

$$\widetilde{\boldsymbol{\theta}} = h_{\boldsymbol{\phi}}(\boldsymbol{\zeta}) \quad \text{with} \quad \boldsymbol{\zeta} \sim p(\boldsymbol{\zeta}) \tag{19}$$

我们，与上面类似，选择先验的$p(\boldsymbol{\zeta})$和函数$h_{\boldsymbol{\phi}}(\boldsymbol{\zeta})$，使得以下成立：

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi};\mathbf{X}) &= \int q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\left(\log p_{\boldsymbol{\theta}}(\mathbf{X}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\right) d\boldsymbol{\theta} \\ &= \int p(\boldsymbol{\zeta})\left(\log p_{\boldsymbol{\theta}}(\mathbf{X}) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta})\right)\Big|_{\boldsymbol{\theta}=h_{\boldsymbol{\phi}}(\boldsymbol{\zeta})} d\boldsymbol{\zeta}\end{aligned} \tag{20}$$

为了表示简洁，我们引入了简写符号$f_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z},\boldsymbol{\theta})$：

$$f_{\boldsymbol{\phi}}(\mathbf{x},\mathbf{z},\boldsymbol{\theta}) = N \cdot (\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z}) + \log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})) + \log p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) - \log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \tag{21}$$

使用等式（20）和（18），给定数据点$\mathbf{x}^{(i)}$的变分下界的蒙特卡罗估计是：

$$\mathcal{L}(\boldsymbol{\phi};\mathbf{X}) \simeq \frac{1}{L}\sum_{l=1}^{L} f_{\boldsymbol{\phi}}(\mathbf{x}^{(l)}, g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(l)},\mathbf{x}^{(l)}), h_{\boldsymbol{\phi}}(\zeta^{(l)})) \tag{22}$$

其中$\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$并且$\zeta^{(l)} \sim p(\boldsymbol{\zeta})$。 估计量仅取决于$p(\boldsymbol{\epsilon})$和$p(\boldsymbol{\zeta})$的样本，这些样本显然不受影响，因此估计量可以通过 w.r.t φ进行区分。得到的随机梯度可以与随机优化方法结合使用，例如 SGD 或 Adagrad [DHS10]。 有关计算随机梯度的基本方法，请参阅算法 1。

F.1 举例

让先验的参数和潜在变量是中心各向同性高斯$p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}) = \mathcal{N}(\mathbf{z};\mathbf{0},\mathbf{I})$和$p_{\boldsymbol{\theta}}(\mathbf{z}) = \mathcal{N}(\mathbf{z};\mathbf{0},\mathbf{I})$。 请注意，在这种情况下，先前缺少参数。 我们还假设真正的后验是近似高斯，具有近似对角线的协方差。 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：

$$\begin{aligned}\log q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) &= \log \mathcal{N}(\boldsymbol{\theta};\boldsymbol{\mu}_{\boldsymbol{\theta}},\boldsymbol{\sigma}_{\boldsymbol{\theta}}^2\mathbf{I}) \\ \log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) &= \log \mathcal{N}(\mathbf{z};\boldsymbol{\mu}_{\mathbf{z}},\boldsymbol{\sigma}_{\mathbf{z}}^2\mathbf{I})\end{aligned} \tag{23}$$

算法 2 用于使用我们的估计器计算随机梯度的伪代码。 有关函数 f_ϕ, g_ϕ 和 h_ϕ 的含义，请参见文本。

要求: ϕ (变分参数的当前值)

g ← 0

for l 从 1 到 L **do:**

x ← 从数据集 X 中随机抽取

ϵ ← 从先验 p(ϵ)中随机抽取

ζ ← 从先验 p(ζ)中随机抽取

$\mathbf{g} \leftarrow \mathbf{g} + \frac{1}{L}\nabla_{\boldsymbol{\phi}} f_{\boldsymbol{\phi}}(\mathbf{x}, g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x}), h_{\boldsymbol{\phi}}(\boldsymbol{\zeta}))$

end for

return g

其中 μ_z 和 $\boldsymbol{\sigma}_{\mathbf{z}}$ 是 x 的未指定函数。 由于它们是高斯型，我们可以参数化变分近似后验：

$$q_{\boldsymbol{\phi}}(\boldsymbol{\theta}) \quad \text{as} \quad \widetilde{\boldsymbol{\theta}} = \boldsymbol{\mu}_{\boldsymbol{\theta}} + \boldsymbol{\sigma}_{\boldsymbol{\theta}} \odot \boldsymbol{\zeta} \qquad \text{where} \quad \boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$
$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \quad \text{as} \quad \widetilde{\mathbf{z}} = \boldsymbol{\mu}_{\mathbf{z}} + \boldsymbol{\sigma}_{\mathbf{z}} \odot \boldsymbol{\epsilon} \qquad \text{where} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

其中 $\odot$ 表示按元素乘积。 这些可以插入上面定义的下限（方程（21）和（22））。

在这种情况下，可以构造具有较低方差的替代估计，因为在该模型中 $p_{\boldsymbol{\alpha}}(\boldsymbol{\theta}), p_{\boldsymbol{\theta}}(\mathbf{z}), q_{\boldsymbol{\phi}}(\boldsymbol{\theta})$ 和 $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ 是高斯的，因此可以解析地求解 f_ϕ 的四个项。 由此产生的估算是：

$$\mathcal{L}(\boldsymbol{\phi}; \mathbf{X}) \simeq \frac{1}{L}\sum_{l=1}^{L} N \cdot \left(\frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\sigma_{\mathbf{z},j}^{(l)})^2) - (\mu_{\mathbf{z},j}^{(l)})^2 - (\sigma_{\mathbf{z},j}^{(l)})^2\right) + \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}\mathbf{z}^{(i)}) \right)$$
$$+ \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\sigma_{\boldsymbol{\theta},j}^{(l)})^2) - (\mu_{\boldsymbol{\theta},j}^{(l)})^2 - (\sigma_{\boldsymbol{\theta},j}^{(l)})^2\right) \tag{24}$$

$\mu_j^{(i)}$ 和 $\sigma_j^{(i)}$ 简单地表示向量 $\boldsymbol{\mu}^{(i)}$ 和 $\boldsymbol{\sigma}^{(i)}$ 的第 j 个元素。