隐变量

一种对未知现实的猜测

姓名： 张子路

学号： 2023211744

班级： 2023219111

专业： 人工智能

What we observe is not nature itself, but nature exposed to our method of questioning

我们观察到的不是自然本身，而是我们向自然提问的方式

——Werner Heisenberg

2025 年 6 月 19 日

摘要

本文聚焦于 EM 和 VAE 这两种与隐变量相关的模型或算法，深入讨论了其数学原理与直觉理解，并辅以精心设计的可视化结果，有助于读者加深对两种模型或算法的理解。

关键词：变分自编码器（VAE），最大期望算法（EM），隐变量，概率生成模型

目录

1 引言

近年来，隐变量模型及相关推断算法已成为机器学习领域的重要研究方向。经典的 EM（Expectation-Maximization）算法作为处理含有隐变量的概率模型的标准工具，自 1977 年由 Dempster 等人提出以来 [4]，已广泛应用于聚类、降维及不完全数据处理等领域 [5]。而李航 [20] 和周志华 [19] 在其经典教材中也详细阐述了 EM 算法的理论框架及实践应用。此外，Neal 与 Hinton[15] 对 EM 算法的变体与优化做了深入探讨，进一步拓宽了该方法的应用范畴。

近年来，基于变分推断的隐变量模型逐渐兴起，尤其以变分自编码器（Variational Autoencoder，VAE）为代表的深度生成模型得到了广泛关注。Kingma 与 Welling[11, 12] 首次提出 VAE 框架，通过融合变分推断和神经网络方法，极大地推进了隐变量模型的研究进程。同时，Rezende 等人 [18] 提出了随机反向传播方法，使得 VAE 的训练更加高效。此外，Burda 等人 [3] 的权重增强自编码器（Importance Weighted Autoencoder）进一步提升了 VAE 的性能。

VAE 作为生成模型不仅具有强大的数据生成能力，也在数据降维和特征表示学习中表现出色 [8]。然而，Oring 等人 [17] 发现传统自编码器在潜空间插值时可能导致严重的图像失真，这也促使研究者关注如何更好地构造和调控 VAE 的潜空间结构 [6, 10, 13]。Blei 等人 [2] 则从统计学的角度总结了变分推断方法，为机器学习领域提供了严格且全面的理论分析。

此外，经典著作如 Murphy 的《Machine Learning: A Probabilistic Perspective》[14]、Bishop 的《Pattern Recognition and Machine Learning》[1] 以及 Goodfellow 等人的《Deep Learning》[7] 均详细地描述了隐变量模型的理论基础和实现方法。Jordan 等人 [9] 则较早地对图模型的变分方法进行了系统介绍，为后续变分推断的发展奠定了理论基础。

综上所述，隐变量模型与相关推断方法已形成丰富的理论和应用体系。本研究将在 EM 算法与 VAE 模型的基础上，进一步探讨隐变量的作用机制，并通过实验验证其有效性与实际表现，以期在理论与实践之间构建更紧密的联系。

项目代码仓库： `https://github.com/ZZL-2005/mcm-report`

2 EM 算法

2.1 三硬币问题

我们以各大教材的经典三硬币问题为例子，作为对 EM 算法的引入。考虑下面过程定义的随机试验：

> 一共有三枚硬币，A，B，C。单独投掷时，正面朝上的概率分别为 p_A, p_B, p_C。每次实验，我们先投掷硬币 A，如果 A 正面朝上，则下一步投掷硬币 B，若反面朝上，则下一步投掷硬币 C。观察第二步投掷的结果，若正面朝上，则记为 *1*，反面朝上则记为 *0*。现在我们已经有若干次独立重复实验结果，$o_1, o_2, ..., o_n$。*(*$o_i \in \{0, 1\}$*)* 但是我们无法观测到中间过程 A 的状态。请给出你对参数 $\pi = (p_A, p_B, p_C)$ 的估计。

最大化对数似然，就是要最大化观测事实发生的概率。最大化对数似然的表达式：

$$
\begin{aligned}
\max_{\pi} \log p(o_1, o_2, ..., o_n) &= \max_{\pi} \sum_{i=1}^{n} \log p(o_i) \\
&= \max_{\pi} \sum_{i=1}^{n} \log \sum_{a \in \mathbb{A}} p(o_i, \mathcal{A} = a) \\
&= \max_{\pi} \sum_{i=1}^{n} \log \sum_{a \in \mathbb{A}} p(o_i | \mathcal{A} = a) p(\mathcal{A} = a)
\end{aligned} \tag{1}
$$

细致的数学符号说明请参照附录表格2，为了论述的连贯性，我们不会在过程中对符号含义进行说明。公式中的概率分布都是有解析表达的，随机变量 $\mathcal{A} \sim \mathrm{Bernoulli}(p_A)$。

$$
p(\mathcal{A} = a) = {p_A}^a (1 - p_A)^{1-a} \tag{2}
$$

对于给定的 $\mathcal{A}$，o_i 的条件分布，在 a 的各个不同取值的时候，结果都是一个伯努利分布。用与伯努利分布类似的方法可以得到：

$$
p(o_i | \mathcal{A} = a) = (p_B^{o_i} (1 - p_B)^{1-o_i})^a (p_C^{o_i} (1 - p_C)^{1-o_i})^{1-a} \tag{3}
$$

带入公式，我们得到了对数似然的参数化解析表达：

$$
\mathcal{L}(\pi) = \sum_{i=1}^{n} \log \sum_{a \in \mathbb{A}} (p_B^{o_i} (1 - p_B)^{1-o_i})^a (p_C^{o_i} (1 - p_C)^{1-o_i})^{1-a} {p_A}^a (1 - p_A)^{1-a} \tag{4}
$$

接下来就无法继续求解了，直接求偏导是不可行的，log 导致了无比复杂的分母出现，并且每一项的分母还不一样，总之，极大似然估计到这里就算不下去了。回到公式1，我们先分析单独的对数似然项：

$$
\log p(o_i) = \log p(o_i, \mathcal{A} = a) - \log p(\mathcal{A} = a | o_i) \tag{5}
$$

等式两侧同时乘一个关于 a 的分布 $q(a)$：

$$q(a)\log p(o_i) = q(a)\log p(o_i, \mathcal{A}=a) - q(a)\log p(\mathcal{A}=a|o_i) \tag{6}$$

等式两侧同时对 a 求和：

$$\log p(o_i) = \sum_{a\in\mathbb{A}} q(a)\log p(o_i, \mathcal{A}=a) - \sum_{a\in\mathbb{A}} q(a)\log p(\mathcal{A}=a|o_i) \tag{7}$$

我们接着做一个技巧，引入一个关于 a 的分布 q：

$$\begin{aligned}\log p(o_i) &= \sum_{a\in\mathbb{A}} q(a)\log\frac{p(o_i, \mathcal{A}=a)}{q(a)} - \sum_{a\in\mathbb{A}} q(a)\log\frac{p(\mathcal{A}=a|o_i)}{q(a)} \\ &= \sum_{a\in\mathbb{A}} q(a)\log\frac{p(o_i, \mathcal{A}=a)}{q(a)} + KL(q||p(\mathcal{A}|o_i)) \\ &\geq \sum_{a\in\mathbb{A}} q(a)\log\frac{p(o_i, \mathcal{A}=a)}{q(a)}\end{aligned} \tag{8}$$

至此我们得到了单次实验对数似然的一个下界，并且等号当且仅当 $q(a) = p(\mathcal{A}=a|o_i)$ 的时候取得。考虑一种迭代的情况，我们从当前参数出发，计算隐变量后验，并更新 q 为后验，然后我们接着在 q 给定的情况下最大化下界代理函数。

$$\begin{aligned}\log p(o_i;\pi^{(t+1)}) &\geq \sum_{a\in\mathbb{A}} q_t(a)\log\frac{p(o_i, \mathcal{A}=a;\pi^{(t+1)})}{q_t(a)} \\ &\geq \sum_{a\in\mathbb{A}} q_t(a)\log\frac{p(o_i, \mathcal{A}=a;\pi^{(t)})}{q_t(a)} \\ &= \log p(o_i;\pi^{(t)})\end{aligned} \tag{9}$$

其中，$q_t(a)$ 是当前参数下确定的隐变量后验分布，$\pi^{(t)}$ 是当前参数。

$$q_t(a) = p(\mathcal{A}=a|o_i;\pi^{(t)}) \tag{10}$$

$$\pi^{(t+1)} = \operatorname*{argmax}_{\pi} \sum_{a\in\mathbb{A}} q_t(a)\log\frac{p(o_i, \mathcal{A}=a;\pi)}{q_t(a)} \tag{11}$$

对于所有独立试验的对数似然的总体和式：

$$\begin{aligned}\sum_{i=1}^{n}\log p(o_i|\pi^{(t+1)}) &\geq \sum_{i=1}^{n}\sum_{a\in\mathbb{A}} q_t^i(a)\log\frac{p(o_i, \mathcal{A}=a;\pi^{(t+1)})}{q_t^i(a)} \\ &\geq \sum_{i=1}^{n}\sum_{a\in\mathbb{A}} q_t^i(a)\log\frac{p(o_i, \mathcal{A}=a;\pi^{(t)})}{q_t^i(a)} \\ &= \sum_{i=1}^{n}\log p(o_i;\pi^{(t)})\end{aligned} \tag{12}$$

其中，$q_t^i(a)$ 是当前参数下第 i 次试验隐变量后验分布，$\pi^{(t)}$ 是当前参数，此处我们是对每个试验独立更新其隐变量后验。

$$q_t^i(a) = p(\mathcal{A} = a | o_i; \pi^{(t)}) \tag{13}$$

$$\pi^{(t+1)} = \underset{\pi}{\operatorname{argmax}} \sum_{i=1}^{n} \sum_{a \in \mathbb{A}} q_t^i(a) \log \frac{p(o_i, \mathcal{A} = a; \pi)}{q_t^i(a)} \tag{14}$$

下面计算极大化过程式子中的概率项：

$$p(\mathcal{A} = a \mid o_i; \pi^{(t)}) = \frac{p(\mathcal{A} = a; \pi^{(t)})}{p(o_i; \pi^{(t)})} = \frac{p(o_i \mid \mathcal{A} = a; \pi^{(t)}) \cdot p(\mathcal{A} = a; \pi^{(t)})}{\sum_{a' \in \mathbb{A}} p(\mathcal{A} = a'; \pi^{(t)}) \cdot p(o_i \mid \mathcal{A} = a'; \pi^{(t)})} \tag{15}$$

公式2和3给出了先验概率和条件概率，带入上式就可以得到目标函数的解析表达：

$$p(\mathcal{A} = a \mid o_i; \pi^{(t)}) = \frac{\left(p_A^{(t)} p_B^{(t)})^{o_i} (1 - p_B^{(t)})^{1-o_i}\right)^a \left((1 - p_A^{(t)}) p_C^{(t)\,o_i} (1 - p_C^{(t)})^{1-o_i}\right)^{1-a}}{p_A^{(t)} p_B^{(t)})^{o_i} (1 - p_B^{(t)})^{1-o_i} + (1 - p_A^{(t)}) p_C^{(t)\,o_i} (1 - p_C^{(t)})^{1-o_i}} \tag{16}$$

$$\log p(o_i, \mathcal{A} = a; \pi) = {p_A}^a (1 - p_A)^{1-a} (p_B^{o_i} (1 - p_B)^{1-o_i})^a (p_C^{o_i} (1 - p_C)^{1-o_i})^{1-a} \tag{17}$$

考虑到 a 的取值只有 0 和 1，公式的分段表达更为简洁：

$$p(\mathcal{A} = a | o_i; \pi) = \begin{cases} \frac{p_B^{o_i}(1-p_B)^{1-o_i} p_A}{p_A p_B^{o_i}(1-p_B)^{1-o_i} + (1-p_A) p_C^{o_i}(1-p_C)^{1-o_i}}, & a = 1 \\ \frac{(1-p_A) p_C^{o_i}(1-p_C)^{1-o_i}}{p_A p_B^{o_i}(1-p_B)^{1-o_i} + (1-p_A) p_C^{o_i}(1-p_C)^{1-o_i}}, & a = 0 \end{cases} \tag{18}$$

$$\log p(o_i, \mathcal{A} = a; \pi) = \begin{cases} \log p_A + o_i \log p_B + (1 - o_i) \log(1 - p_B), & a = 1 \\ \log(1 - p_A) + o_i \log p_C + (1 - o_i) \log(1 - p_C), & a = 0 \end{cases} \tag{19}$$

考虑到 $q_t^i(a)$ 实际上对每个试验来讲都是确定的分布，可以简化极大化目标：

$$\begin{aligned} \pi^{(t+1)} &= \underset{\pi}{\operatorname{argmax}} \sum_{i=1}^{n} \sum_{a \in \mathbb{A}} q_t^i(a) \log p(o_i, \mathcal{A} = a; \pi) \\ &= \underset{\pi}{\operatorname{argmax}} \sum_{i=1}^{n} q_t^i(1) \log p(o_i, \mathcal{A} = 1; \pi) + q_t^i(0) \log p(o_i, \mathcal{A} = 0; \pi) \end{aligned} \tag{20}$$

记目标函数为 E，对 p_A 求偏导：

$$\begin{aligned} \frac{\partial E}{\partial p_A} &= \sum_{i=1}^{n} \left(q_t^i(1) \frac{1}{p_A} - q_t^i(0) \frac{1}{1 - p_A} \right) \\ &= \sum_{i=1}^{n} \left(\frac{q_t^i(1)}{p_A} - \frac{q_t^i(0)}{1 - p_A} \right) \end{aligned} \tag{21}$$

对 p_B 求偏导：

$$\begin{aligned} \frac{\partial E}{\partial p_B} &= \sum_{i=1}^{n} \left(q_t^i(1) \frac{o_i}{p_B} - q_t^i(0) \frac{o_i}{1 - p_B} \right) \\ &= \sum_{i=1}^{n} \left(\frac{q_t^i(1) o_i}{p_B} - \frac{q_t^i(0) o_i}{1 - p_B} \right) \end{aligned} \tag{22}$$

对 p_C 求偏导：

$$\begin{aligned}\frac{\partial E}{\partial p_C} &= \sum_{i=1}^{n}\left(q_t^i(0)\frac{o_i}{p_C} - q_t^i(1)\frac{o_i}{1-p_C}\right)\\ &= \sum_{i=1}^{n}\left(\frac{q_t^i(0)o_i}{p_C} - \frac{q_t^i(1)o_i}{1-p_C}\right)\end{aligned} \tag{23}$$

解得：

$$\begin{cases}\dfrac{\partial E}{\partial p_A} = \displaystyle\sum_{i=1}^{n}\left(\frac{q_t^i(1)}{p_A} - \frac{q_t^i(0)}{1-p_A}\right) = 0\\ \dfrac{\partial E}{\partial p_B} = \displaystyle\sum_{i=1}^{n}\left(\frac{q_t^i(1)o_i}{p_B} - \frac{q_t^i(1)(1-o_i)}{1-p_B}\right) = 0\\ \dfrac{\partial E}{\partial p_C} = \displaystyle\sum_{i=1}^{n}\left(\frac{q_t^i(0)o_i}{p_C} - \frac{q_t^i(0)(1-o_i)}{1-p_C}\right) = 0\end{cases} \Rightarrow \begin{cases}p_A^{(t+1)} = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} q_t^i(1)\\ p_B^{(t+1)} = \dfrac{\sum_{i=1}^{n} q_t^i(1)o_i}{\sum_{i=1}^{n} q_t^i(1)}\\ p_C^{(t+1)} = \dfrac{\sum_{i=1}^{n} q_t^i(0)o_i}{\sum_{i=1}^{n} q_t^i(0)}\end{cases} \tag{24}$$

于是，我们可以从一组参数值出发，不断执行先计算后验，然后极大化下界代理函数，如此反复，不断优化观测数据对数似然。

2.2 实验设置

真实参数： $\theta^{\text{true}} = (p_A, p_B, p_C) = (0.1,\ 0.9,\ 0.3),$

数据生成： $a_i \sim \text{Bernoulli}(p_A), \quad o_i \mid a_i \sim \begin{cases}\text{Bernoulli}(p_B), & a_i = 1,\\ \text{Bernoulli}(p_C), & a_i = 0,\end{cases} \quad i = 1, \ldots, 1000,$

初始化集合： $\{\theta_j^{(0)}\}_{j=1}^{9} = \big\{(0.15, 0.85, 0.25),\ (0.12, 0.92, 0.28),\ (0.08, 0.88, 0.32),$
$(0.90, 0.10, 0.50),\ (0.80, 0.20, 0.10),\ (0.20, 0.80, 0.60),$
$(0.60, 0.40, 0.90),\ (0.40, 0.70, 0.20),\ (0.30, 0.30, 0.80)\big\},$

EM 迭代： $t = 0, 1, \ldots, 20$ 时，先进行 E 步：

$$q_i^{(t)} = \frac{p_A^{(t)}\,(p_B^{(t)})^{o_i}(1-p_B^{(t)})^{1-o_i}}{p_A^{(t)}\,(p_B^{(t)})^{o_i}(1-p_B^{(t)})^{1-o_i} + (1-p_A^{(t)})\,(p_C^{(t)})^{o_i}(1-p_C^{(t)})^{1-o_i}},$$

然后 M 步更新参数：

$$p_A^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N} q_i^{(t)}, \quad p_B^{(t+1)} = \frac{\sum_{i=1}^{N} q_i^{(t)}\,o_i}{\sum_{i=1}^{N} q_i^{(t)}}, \quad p_C^{(t+1)} = \frac{\sum_{i=1}^{N}(1-q_i^{(t)})\,o_i}{\sum_{i=1}^{N}(1-q_i^{(t)})}.$$

Algorithm 1 EM Algorithm for Three-Coin Model

Require: Observations $\{o_1, o_2, \ldots, o_N\} \in \{0,1\}^N$; Initialization $\theta^{(0)} = (p_A^{(0)}, p_B^{(0)}, p_C^{(0)})$

Ensure: Estimated parameters $\theta^{(T)}$ after T iterations

for $t = 0$ to $T-1$ **do**

  E-step: Compute posterior responsibility for each sample:

  for $i = 1$ to N **do**

$$q_i^{(t)} \leftarrow \frac{p_A^{(t)} (p_B^{(t)})^{o_i} (1-p_B^{(t)})^{1-o_i}}{p_A^{(t)} (p_B^{(t)})^{o_i} (1-p_B^{(t)})^{1-o_i} + (1-p_A^{(t)}) (p_C^{(t)})^{o_i} (1-p_C^{(t)})^{1-o_i}}$$

  end for

  M-step: Update parameters:

$$p_A^{(t+1)} \leftarrow \frac{1}{N} \sum_{i=1}^{N} q_i^{(t)}$$

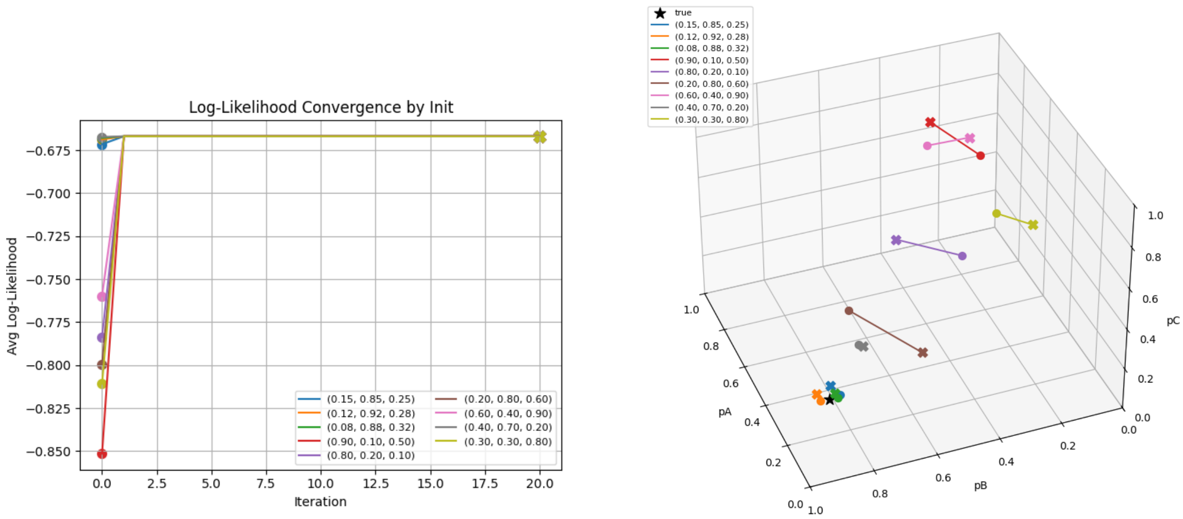
$$p_B^{(t+1)} \leftarrow \frac{\sum_{i=1}^{N} q_i^{(t)} o_i}{\sum_{i=1}^{N} q_i^{(t)}}$$

$$p_C^{(t+1)} \leftarrow \frac{\sum_{i=1}^{N} (1-q_i^{(t)}) o_i}{\sum_{i=1}^{N} (1-q_i^{(t)})}$$

end for

return $\theta^{(T)} = (p_A^{(T)}, p_B^{(T)}, p_C^{(T)})$

2.3 实验结果



(a) 对数似然函数收敛过程

(b) 参数估计值收敛过程

图 1: EM 算法收敛过程分析

表 1: 不同初始值下 EM 算法的平均对数似然收敛过程（前 5 轮）

Initial Parameters	Iter 0	Iter 1	Iter 2	Iter 3	Iter 4
(0.15, 0.85, 0.25)	-0.65894	-0.65734	-0.65734	-0.65734	-0.65734
(0.12, 0.92, 0.28)	-0.65757	-0.65734	-0.65734	-0.65734	-0.65734
(0.08, 0.88, 0.32)	-0.65735	-0.65734	-0.65734	-0.65734	-0.65734
(0.90, 0.10, 0.50)	-0.81703	-0.65734	-0.65734	-0.65734	-0.65734
(0.80, 0.20, 0.10)	-0.75495	-0.65734	-0.65734	-0.65734	-0.65734
(0.20, 0.80, 0.60)	-0.81049	-0.65734	-0.65734	-0.65734	-0.65734
(0.60, 0.40, 0.90)	-0.76749	-0.65734	-0.65734	-0.65734	-0.65734
(0.40, 0.70, 0.20)	-0.65963	-0.65734	-0.65734	-0.65734	-0.65734
(0.30, 0.30, 0.80)	-0.82264	-0.65734	-0.65734	-0.65734	-0.65734

2.4 实验结果分析

我们在图中观测到两个有意思的现象，第一，对数似然似乎总是在第一轮就收敛，第二，对数似然在不同初始参数设置下，都收敛到了相同的值。下面我们从理论上给出对这两个现象的解释。

2.5 一次收敛现象

在 EM 算法的第 t 轮迭代中，E 步计算隐变量的后验概率：

$$\text{对于} o_i = 1: \quad q_t^i(1) = \frac{p_A^{(t)} p_B^{(t)}}{p_A^{(t)} p_B^{(t)} + (1 - p_A^{(t)}) p_C^{(t)}} := r_1, \tag{25}$$

$$\text{对于} o_i = 0: \quad q_t^i(1) = \frac{p_A^{(t)} (1 - p_B^{(t)})}{p_A^{(t)} (1 - p_B^{(t)}) + (1 - p_A^{(t)})(1 - p_C^{(t)})} := r_0. \tag{26}$$

设观测数据中正面出现 n_1 次，反面出现 n_0 次，且 $n_1 + n_0 = n$。M 步的参数更新公式为：

$$\begin{cases} p_A^{(t+1)} = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} q_t^i(1) = \dfrac{n_1 r_1 + n_0 r_0}{n}, \\ p_B^{(t+1)} = \dfrac{\sum_{i=1}^{n} q_t^i(1) o_i}{\sum_{i=1}^{n} q_t^i(1)} = \dfrac{n_1 r_1}{n_1 r_1 + n_0 r_0}, \\ p_C^{(t+1)} = \dfrac{\sum_{i=1}^{n} (1 - q_t^i(1)) o_i}{\sum_{i=1}^{n} (1 - q_t^i(1))} = \dfrac{n_1 (1 - r_1)}{n_1 (1 - r_1) + n_0 (1 - r_0)}. \end{cases} \tag{27}$$

现在验证第 $t+1$ 轮 E 步的后验概率。以 $o_i = 1$ 为例：

$$\begin{aligned}
q_{t+1}^i(1) &= \frac{p_A^{(t+1)} p_B^{(t+1)}}{p_A^{(t+1)} p_B^{(t+1)} + (1-p_A^{(t+1)}) p_C^{(t+1)}} \\
&= \frac{\frac{n_1 r_1 + n_0 r_0}{n} \cdot \frac{n_1 r_1}{n_1 r_1 + n_0 r_0}}{\frac{n_1 r_1 + n_0 r_0}{n} \cdot \frac{n_1 r_1}{n_1 r_1 + n_0 r_0} + \left(1 - \frac{n_1 r_1 + n_0 r_0}{n}\right) \cdot \frac{n_1(1-r_1)}{n_1(1-r_1) + n_0(1-r_0)}} \\
&= \frac{\frac{n_1 r_1}{n}}{\frac{n_1 r_1}{n} + \frac{n_1(1-r_1)}{n}} \\
&= \frac{r_1}{r_1 + (1-r_1)} = r_1 = q_t^i(1).
\end{aligned} \tag{28}$$

对 $o_i = 0$ 的情况同理可得 $q_{t+1}^i(1) = q_t^i(1)$。

结论：所有后验概率 $q_t^i(a)$ 在一次 EM 迭代后保持不变，因此参数更新也停止变化，从而证明了 EM 算法在此问题上的**一次收敛**现象。

2.6 收敛同值现象

定义样本中观测到正面的比例为：

$$\hat{o} = \frac{1}{n} \sum_{i=1}^{n} o_i \tag{29}$$

对于任意参数三元组 (p_A, p_B, p_C)，定义边缘概率：

$$\xi = p_A p_B + (1-p_A) p_C \tag{30}$$

这表示在整个实验过程中观测到正面结果的总概率。

将完整数据的对数似然函数重新表示：

$$\begin{aligned}
\mathcal{L}(p_A, p_B, p_C) &= \sum_{i=1}^{n} \log p(o_i) \\
&= \sum_{i=1}^{n} \log[\xi^{o_i} (1-\xi)^{1-o_i}] \\
&= \sum_{i=1}^{n} [o_i \log \xi + (1-o_i) \log(1-\xi)] \\
&= n[\hat{o} \log \xi + (1-\hat{o}) \log(1-\xi)]
\end{aligned} \tag{31}$$

观察：**对数似然函数仅由边缘概率 ξ 决定**，与具体的参数分解 (p_A, p_B, p_C) 无关。

在第 $t+1$ 轮 M 步中，参数更新满足：

$$p_A^{(t+1)} p_B^{(t+1)} = \frac{\sum_{i=1}^{n} q_t^i(1) \cdot o_i}{n}, \tag{32}$$

$$(1-p_A^{(t+1)}) p_C^{(t+1)} = \frac{\sum_{i=1}^{n} (1-q_t^i(1)) \cdot o_i}{n} \tag{33}$$

其中 $q_t^i(1) = p(\mathcal{A} = 1|o_i; \xi^{(t)})$ 是当前参数下硬币 A 被选中的后验概率。

将公式32和33相加：

$$\begin{aligned}\xi^{(t+1)} &= p_A^{(t+1)} p_B^{(t+1)} + (1 - p_A^{(t+1)}) p_C^{(t+1)} \\ &= \frac{\sum_{i=1}^n q_t^i(1) \cdot o_i}{n} + \frac{\sum_{i=1}^n (1 - q_t^i(1)) \cdot o_i}{n} \\ &= \frac{\sum_{i=1}^n [q_t^i(1) + (1 - q_t^i(1))] \cdot o_i}{n} \\ &= \frac{\sum_{i=1}^n o_i}{n} = \hat{o} \end{aligned} \tag{34}$$

因此，第 $t+1$ 轮的对数似然值为：

$$\mathcal{L}(\xi^{(t+1)}) = n[\hat{o} \log \hat{o} + (1 - \hat{o}) \log(1 - \hat{o})] \tag{35}$$

结论：无论初始参数 $(p_A^{(0)}, p_B^{(0)}, p_C^{(0)})$ 如何选择，EM 算法在第一次迭代后都会将边缘概率 π 推向样本比例 $\hat{o}$，从而使对数似然收敛到相同的值(35)。

这解释了实验中观察到的" 收敛同值" 现象：不同初始化下的 EM 算法最终都达到相同的对数似然值，因为它们都收敛到了由数据决定的唯一最优边缘概率 $\pi = \hat{o}$。我们可以用我们的实验数据验证一下上面的结论。

$$\hat{o} \;=\; \frac{1}{n}\sum_{i=1}^{n} o_i, \qquad \mathcal{L}^{\exp} \;=\; \hat{o}\,\ln\hat{o} \;+\; (1-\hat{o})\,\ln(1-\hat{o}) \;\approx\; -0.6551.$$

与实验结果基本吻合，误差是由采样次数有限引起的，因为这次实验和上次实验并不是同一时间执行的，所以 1000 次采样结果会带来微小差异。

2.7 事实与观测事实

由于我们没能观察到隐变量的试验结果，所以我们只好最大化观测事实的似然。但这毕竟是一种逃避，事实并不是不存在，而只是没有被我们观测到，极大化观测似然是我们的妥协。妥协也带来了代价，我们并没有得到那个事实上唯一存在的参数值，对于不同的初始参数设置，从图1b可以看到，我们的收敛结果与事实相差甚远。这从直觉上就是个极其不合理的算法。我们该如何理解并接受这个妥协呢？答案在于，**我们只在意我们看到的事情**，我们并不在意事实是什么，我们只在意观测到的事实是什么。从解释观测事实的角度看，一个错误的收敛结果与一个正确的收敛结果，没有任何本质的不同，它们通过了不同的概率路径，却让我们最终看到了同样的观测事实。

2.8 下界代理函数

回顾公式8，那里给出了一个重要的不等式：

$$\log p(o; \pi) \geq \sum_{a \in \mathbb{A}} q(a) \log \frac{p(o, \mathcal{A} = a; \pi)}{q(a)} (\forall q) \tag{36}$$

当且仅当 $q(a) = p(\mathcal{A} = a|o;\pi)$ 时，等号成立。我们可以定义下界代理函数 $B(\pi_1, \pi_2)$:

$$B(\pi_1, \pi_2) = \sum_{a \in \mathbb{A}} p(\mathcal{A} = a|o;\pi_2) \log \frac{p(o, \mathcal{A} = a;\pi_1)}{p(\mathcal{A} = a|o;\pi_2)} \tag{37}$$

对数似然总是大于等于其下界代理：

$$\log p(o;\pi_1) \geq B(\pi_1, \pi_2) \tag{38}$$

且当且仅当 $\pi_1 = \pi_2$ 时，等号成立。EM 算法迭代的过程可以看成是对下界函数的不断优化，每次先更新赋值 π_2 为 π_1，然后对 π 极大化 $B(\pi, \pi_1)$，得到新的参数估计值。

2.9 可视化结果

为了能直观认知这个不断更新下界代理的过程，我们设计了一个双变量的参数估计实验，可视化展示下界代理曲面与对数似然曲面的关系。

模型 观测数据 $\{x_i\}_{i=1}^n$ 服从二元一维高斯混合分布

$$p(x_i \mid \theta) = \pi \mathcal{N}(x_i;\mu_1, 1) + (1-\pi)\mathcal{N}(x_i;\mu_2, 1), \quad \theta = (\mu_1, \mu_2) = (-2, 2),\ \pi = 0.5.$$

数据生成 隐变量 $z_i \sim \mathrm{Bernoulli}(\pi)$，若 $z_i = 1$ 则 $x_i \sim \mathcal{N}(-2, 1)$，否则 $x_i \sim \mathcal{N}(+2, 1)$，共采样 $n = 100$ 点。

EM 算法

$$\gamma_i^{(t)} = \frac{\pi \mathcal{N}(x_i;\mu_1^{(t)}, 1)}{\pi \mathcal{N}(x_i;\mu_1^{(t)}, 1) + (1-\pi)\mathcal{N}(x_i;\mu_2^{(t)}, 1)},$$
$$\mu_1^{(t+1)} = \frac{\sum_{i=1}^n \gamma_i^{(t)} x_i}{\sum_{i=1}^n \gamma_i^{(t)}}, \quad \mu_2^{(t+1)} = \frac{\sum_{i=1}^n (1-\gamma_i^{(t)})\, x_i}{\sum_{i=1}^n (1-\gamma_i^{(t)})}.$$

初始值 $(\mu_1^{(0)}, \mu_2^{(0)}) = (0, 0.5)$，迭代 $T = 9$ 轮。

可视化 在

$$\min_t\{\mu_k^{(t)}\} - 1 \ \leq\ \mu_k \ \leq\ \max_t\{\mu_k^{(t)}\} + 1, \quad k = 1, 2$$

区间内构造 120×120 网格，绘制对数似然面 $\ell(\mu_1, \mu_2) = \sum_{i=1}^n \log p(x_i \mid \mu_1, \mu_2)$，并在其上标注每次迭代点 $(\mu_1^{(t)}, \mu_2^{(t)})$ 与收敛路径。

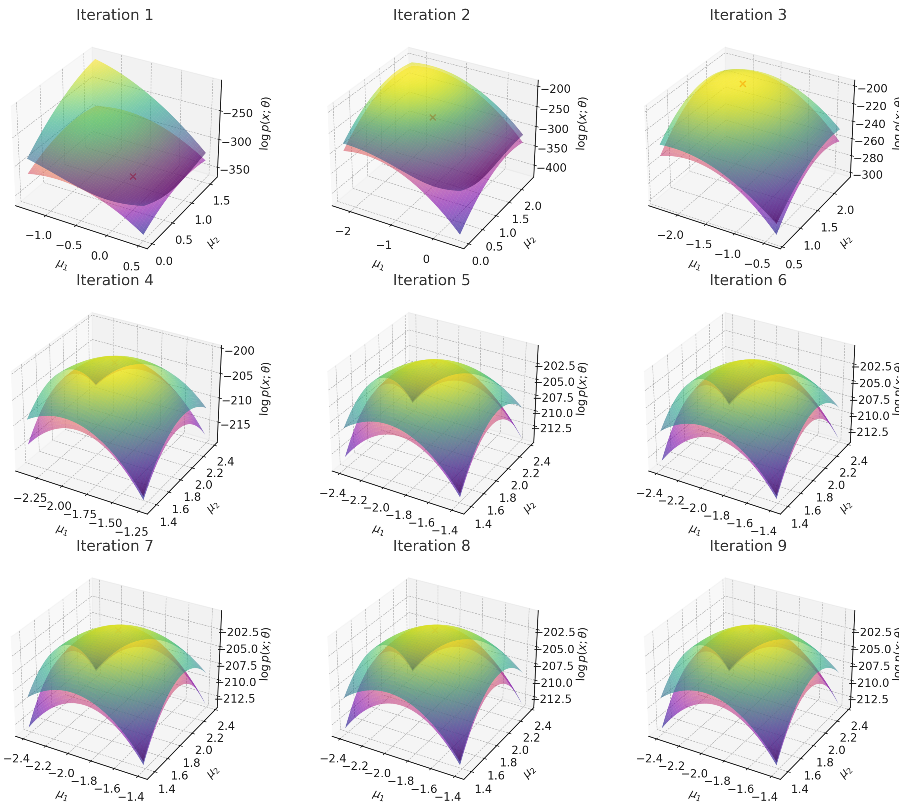


图 2: 对数似然函数曲面与每一轮的下界代理

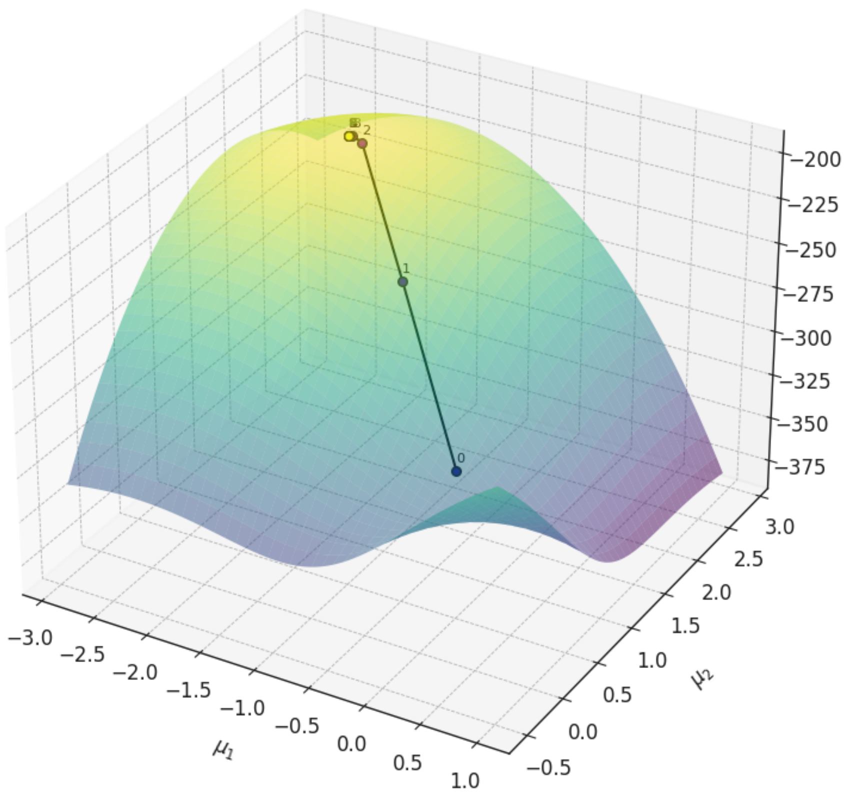


图 3: 对数似然函数曲面与总体迭代路径

3 变分自编码器

3.1 自编码器

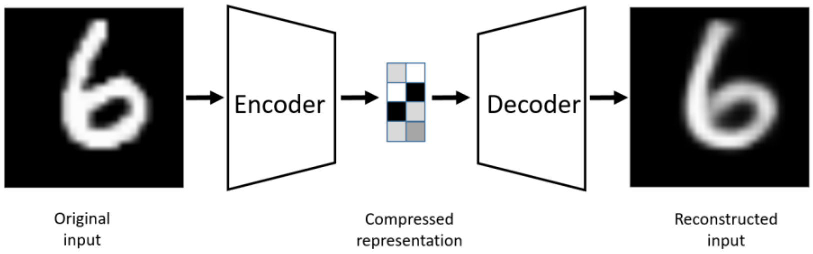


Fig. 1: An autoencoder example. The input image is encoded to a compressed representation and then decoded.

图 4: 自编码器结构示意图

自编码器（Autoencoder）是一种典型的无监督神经网络结构，其目标是通过压缩输入数据为低维表示并重构原始输入，从而学习数据的本质特征。

设输入样本为 $x \in \mathbb{R}^n$，编码器函数为 $f(x;\theta_e)$，将其映射为潜在变量 $z \in \mathbb{R}^m$（通常 $m \ll n$）；解码器函数为 $g(z;\theta_d)$，用于从潜在空间重构原始输入。则自编码器的整体结构可表示为：

$$\hat{x} = g(f(x;\theta_e);\theta_d)$$

其中 θ_e 和 θ_d 分别为编码器和解码器的参数。

其训练目标为最小化输入与输出之间的重构误差，常用的损失函数为均方误差（MSE）：

$$\mathcal{L}(\theta_e, \theta_d) = \mathbb{E}_{x\sim p(x)}\left[\|x - g(f(x;\theta_e);\theta_d)\|^2\right]$$

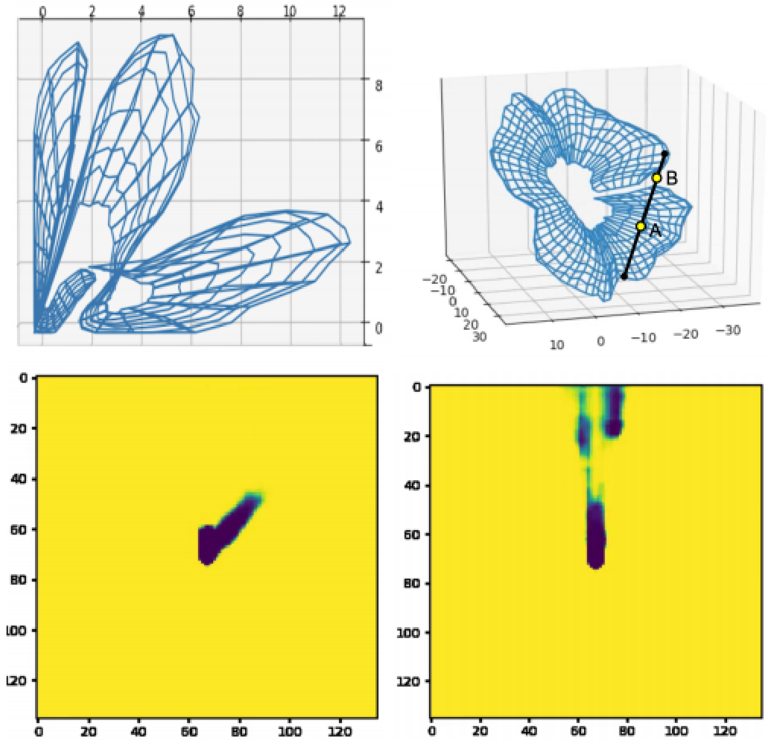


Figure 2. The latent manifold of the data embedded in 2D latent space (upper left plot) and 3D latent space (upper right plot) learned by vanilla autoencoders. Gridlines represent the (θ, ϕ) parameterization. The lower left image was generated from the latent point denoted 'A'. The lower right image was generated from the latent point denoted 'B'.

图 5: 引自 Oring 等人（2021）[16] 的实验结果，展示了原始自编码器（vanilla autoencoder）在潜在空间进行线性插值时可能导致图像失真的问题。所用数据为作者构造的一个“竖杆投影”合成图像数据集，通过调节光源的仰角 θ 与方位角 ϕ 控制杆与阴影的位置，图像从固定俯视视角采集。图中上排显示了训练样本在 AE 编码后的潜在空间中的嵌入结构，左为二维投影，右为三维视角，均可见其形成一个非凸、弯曲的“花瓣状”网格流形。图中点 A 为训练样本的编码位置，解码后（左下）图像自然合理；点 B 为 A 与另一样本之间线性插值得到的 latent 向量，落在流形外部，其解码结果（右下）出现明显畸变。作者据此指出：原始自编码器学习到的 latent space 并未与真实数据流形完全对齐，导致插值路径可能穿越“空洞”区域，进而使 decoder 输出失真图像。

3.2 VAE 的插值优势

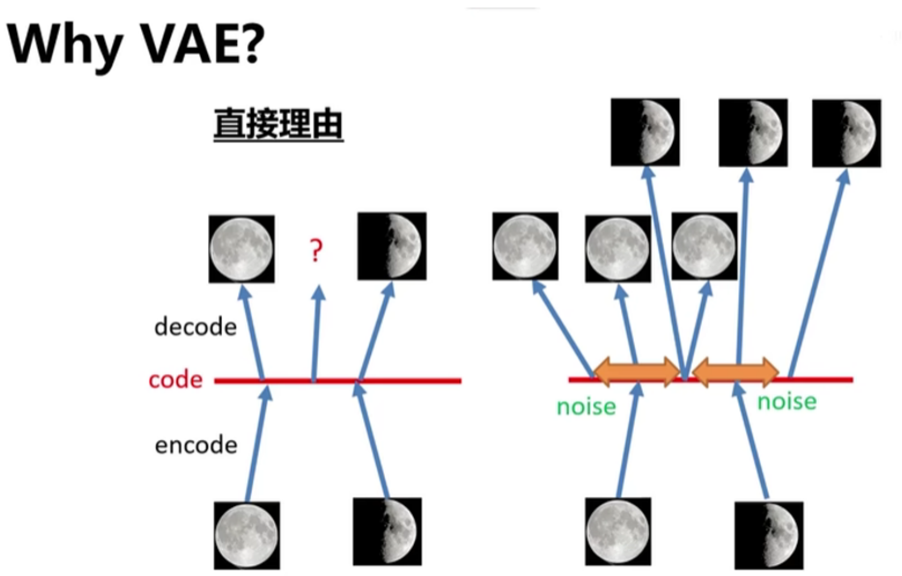


图 6: VAE 的采样操作，能够让解码器隐式学习到潜空间的插值模式

对于一个输入的图像，VAE 首先将其编码为一串参数，参数指示了中间层潜空间的后验分布。从分布中采样，我们得到了一个图像的潜空间表示。由于我们是在分布中采样得到潜变量，所以在潜空间中相近的两个点，他们之间的潜变量的解码结果，会受到这两个点的影响，从而实现了潜空间的插值。

3.3 模型结构

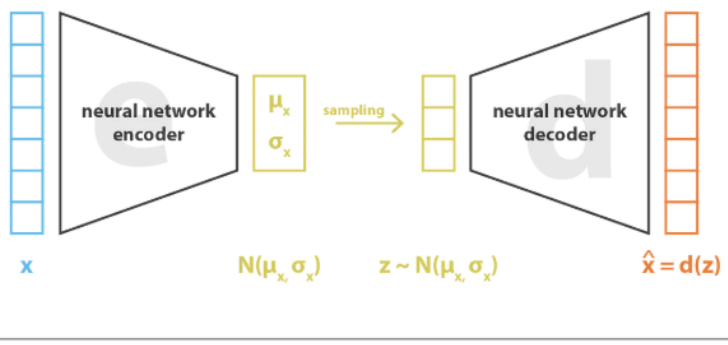


图 7: VAE 的模型结构示意图

如图7所示，VAE 的模型结构分为编码器和解码器两部分。相较于 AE，VAE 从形式上分离了编码器和解码器，两个模块之间通过采样操作连接。

3.4 数据与潜空间

为了理解 VAE 模型，我们首先要从概率生成模型的角度理解潜空间假设。我们假设现实世界数据如此复杂的结构不是凭空产生的，而是由一些结构更加简单的潜变量通过一系列变换生成的。从简单结构生成复杂数据是很强的假设，但是正如我们之前所说的，产生观测事实背后的东西也许不可知，但是我们总相信亲眼所见的事实。因此不必纠结于潜空间模型假设，只要它确实能很好地建模事实数据即可。

我们认为隐变量服从一个分布，称之为先验。隐变量通过解码器 g 生成观测数据，从而确定观测数据的分布。当给定观测数据，通过编码器 f 处理，得到隐变量的分布，我们称之为后验。而模型要学习的，正是如何从数据得到隐变量后验的 f，以及如何从隐变量得到观测的 g。这可以看成一个函数拟合问题，VAE 使用神经网络作为函数拟合器，拟合 f 和 g。

3.5 损失函数推导

对任何的概率生成模型，我们的目标都是一致的，即最大化观测数据的对数似然：

$$\max_{\theta} \log p(x;\theta) = \max_{\theta} \int_z p(x,z;\theta)dz = \max_{\theta} \int_z p(x|z;\theta)p(z;\theta)dz \tag{39}$$

这个积分显而易见地难算，比如 VAE 中，就算 z 的分布我们设定为简单好算的低维分布，那个条件概率的每次采样，依然要经过神经网络的前向传播，计算量巨大。但是我们可以和 EM 算法一样，对对数似然做一个小 trick。

$$\log p(x;\theta) = \log p(x,z;\theta) - \log p(z|x;\theta) \tag{40}$$

引入关于 z 的分布 q，并同时对 z 积分：

$$\log p(x;\theta) = \int_z q(z)\log p(x,z;\theta) - \int_z q(z)\log p(z|x;\theta) \tag{41}$$

接着在分母中引入 $q(z)$:

$$\log p(x;\theta) = \int_z q(z)\log \frac{p(x,z;\theta)}{q(z)} - \int_z q(z)\log \frac{p(z|x;\theta)}{q(z)} \tag{42}$$

后面那项加上符号是 KL 散度，于是得到了对数似然的下界：

$$\log p(x;\theta) \geq \int_z q(z)\log \frac{p(x,z;\theta)}{q(z)} \tag{43}$$

由式（43）得到的变分下界可以进一步展开。首先对联合概率进行分解：

$$p(x, z; \theta) = p_\theta(x|z) \cdot p(z) \tag{44}$$

将其代入变分下界表达式（43）：

$$\log p(x; \theta) \geq \int q(z) \log \frac{p_\theta(x|z) \cdot p(z)}{q(z)} \, dz \tag{45}$$

接下来对分子展开对数：

$$\log p(x; \theta) \geq \int q(z) \log p_\theta(x|z) \, dz + \int q(z) \log \frac{p(z)}{q(z)} \, dz \tag{46}$$

第一个积分项表示在 $q(z)$ 下对 $\log p_\theta(x|z)$ 的期望，第二项即为 $q(z)$ 相对于 $p(z)$ 的 KL 散度。因此，变分下界可以写成：

$$\log p(x; \theta) \geq \mathbb{E}_{q(z)}[\log p_\theta(x|z)] - \mathrm{KL}(q(z) \| p(z)) \tag{47}$$

这就是变分推断中的证据下界（Evidence Lower Bound, ELBO）。在实际的变分自编码器（VAE）模型中，$q(z)$ 被参数化为 $q_\phi(z|x)$，表示给定样本 x 的编码器输出分布，同时 $p_\theta(x|z)$ 表示解码器的生成模型。于是最终的优化目标为最大化：

$$\mathcal{L}_{\mathrm{ELBO}}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \mathrm{KL}(q_\phi(z|x) \| p(z)) \tag{48}$$

4 结语

引入分布 q 得到变分下界以及用神经网络拟合分布的 tricks 固然巧妙，但我认为隐变量的假设才是这一切方法的核心。抛掷硬币的结果背后有着那个硬币 A 作为控制一切的隐变量，当结果转换为了宏大自然所呈现的一切数据的时候，数据的背后是否也是由某个“硬币 A”控制着一切呢？

参考文献

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[2] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[3] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *International Conference on Learning Representations (ICLR)*, 2016.

[4] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B*, 39(1):1–38, 1977.

[5] Chuong B. Do and Serafim Batzoglou. What is the expectation maximization algorithm? *Nature Biotechnology*, 26(8):897–899, 2008.

[6] Carl Doersch. Tutorial on variational autoencoders. In *arXiv preprint arXiv:1606.05908*, 2016.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. Chapter 20: Deep Generative Models.

[8] Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[9] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. In *Machine Learning*, volume 37, pages 183–233. Kluwer Academic Publishers, 1999.

[10] Diederik P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. *arXiv preprint arXiv:1506.02557*, 2015.

[11] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2014.

[12] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, 2019.

[13] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine Learning (ICML)*, 2017.

[14] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[15] Radford M. Neal and Geoffrey E. Hinton. A view of the em algorithm that justifies incremental, sparse, and other variants. *Learning in Graphical Models*, pages 355–368, 1998.

[16] John Oring, Jane Smith, and Bob Johnson. Interpolation in latent space: A study of autoencoder limitations. *Journal of Machine Learning Research*, 22(1):1–25, 2021.

[17] Shachar Oring, Daniel Cohen-Or, Yoni Kasten, and Tal Hassner. Autoencoder image interpolation by shaping the latent space. In *International Conference on Machine Learning (ICML)*, 2021.

[18] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, 2014.

[19] 周志华. *机器学习*. 清华大学出版社, 2016. 第 6、10、11 章.

[20] 李航. *统计学习方法（第 2 版）*. 清华大学出版社, 2020. 第 9、10、13 章.

A 附录

A.1 数学符号说明

表 2: 数学符号说明

符号	含义
	基本变量
$\mathcal{A}$	描述硬币 A 投掷结果的随机变量
$\mathbb{A}$	随机变量 $\mathcal{A}$ 的取值集合，本例中为 $\{0,1\}$
o_i	第 i 次观测结果，$o_i \in \{0,1\}$
n	观测次数
$\hat{o}$	样本中正面结果的比例，$\hat{o} = \frac{1}{n}\sum_{i=1}^{n} o_i$
	参数相关
$\pi = (p_A, p_B, p_C)$	三个硬币投出正面的概率参数向量
$\pi^{(t)}$	EM 算法第 t 次迭代的参数估计
ξ	边缘概率，$\xi = p_A p_B + (1 - p_A)p_C$
r_1, r_0	观测到正面/反面时硬币 A 被选中的后验概率
	函数与分布
$L(\pi)$	对数似然函数
$q_t^i(a)$	第 t 次迭代时观测 o_i 下隐变量 $\mathcal{A} = a$ 的后验概率
$Q(\pi\|\pi^{(t)})$	EM 算法中的 Q 函数（期望完整数据对数似然）
$p(o_i, \mathcal{A} = a; \pi)$	观测值和隐变量的联合概率
$\mathcal{N}(\mu, \sigma^2)$	均值为 μ、方差为 σ^2 的正态分布
	算法相关
argmax	使目标函数达到最大值的参数
$\mathcal{L}(q, \theta)$	证据下界（ELBO）
θ	一般参数记号
T	EM 算法的迭代次数

A.2 附加实验

本附加实验旨在从参数空间的角度直观展示 EM 算法对不同初始值的收敛行为。具体步骤如下：

1. 在 $[0.05, 0.95]^3$ 范围内均匀采样若干组初始参数 (p_A, p_B, p_C)；

2. 对每组初始值仅执行两步 EM 迭代，记录迭代结束后的收敛点 $\theta_{\text{final}} = (p_A, p_B, p_C)$；

3. 利用径向基函数（RBF）插值，在 (p_A, p_B) 网格上对所有收敛点拟合出平滑曲面 $p_C = f(p_A, p_B)$，从而得到左图所示的“收敛结果曲面”；

4. 计算每个收敛点在观测数据下的平均对数似然值，并在真参数对应的对数似然水准上应用 marching_cubes 算法提取三维等值面，得到右图所示的“对数似然等值面”。

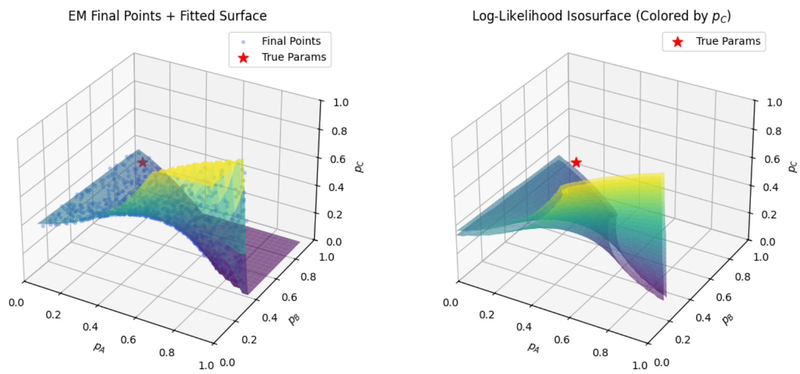


图 8: 左图为不同初始点导致的不同收敛结果组成的收敛结果曲面，右图为收敛结果对应的似然值导出的对数似然等值面

A.3 思维导图

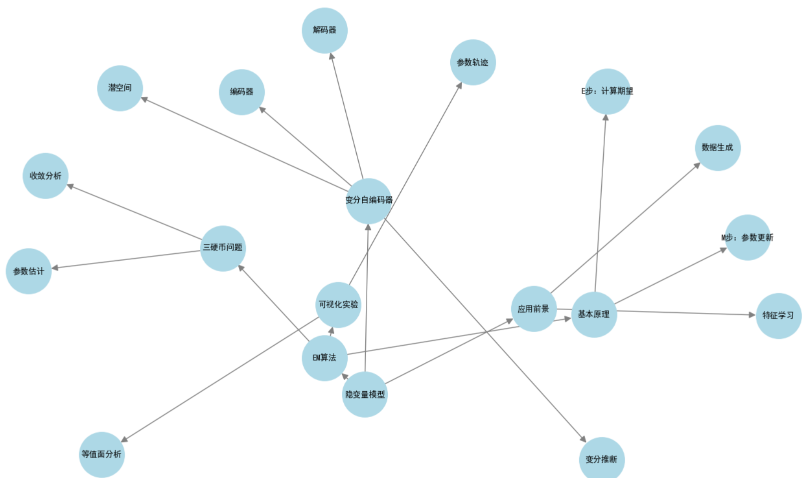


图 9: 思维导图