深度变分贝叶斯滤波器:
基于原始数据的非监督学习状态空间模型
Deep Variational Bayes Filters Unsupervised Learning of State Space Models from Raw Data

翻译：李彦增, 谢江

2018年12月15日

摘要

我们提出了深度变分贝叶斯过滤器(DVBF)，这种新方法可以用于无监督学习，并能识别潜在的马尔科夫状态空间模型。DVBF 利用了随机梯度变分贝叶斯的最新成果，能够使用变分推断之前难以处理的推断分布。因此，它可以处理具有时空依赖性的非线性数据(如没有领域知识的图像序列)。我们的实验表明，通过变换使用反向传播可以实施状态空间假设，并明显增强数据的潜在嵌入(embedding)信息量。同时，它也可以在实际中实现长期预测。

1 简介

针对于序列数据的概率模型估算应用是音频、自然语言、工业植物种植等领域的研究热点[8, 24, 4, 5, 15]。这些研究的最终目标是获得一个能够反映观测序列$X_1 : T$ 数据集的模型$p(X_1 : T)$。近年来，随着深度学习的发展，能够对高维时间依赖性序列数据进行拟合的强大模型开始出现[8, 24, 4, 1]。

在系统理论中，动态系统的时间序列已经得到了充分的研究[18]。尤其是目前已经证明了状态空间模型是解决动力分析与控制学问题的强大工具，但这方面还有两项极具挑战性的任务：我们是否能够仅靠数据确定调节系统(governing system)？我们是否可以从可观察的数据推断出系统隐变量？这两个任务是相对的：更强大的系统表示需要更高算力的推断支持，但更有效的推断[例如着名的卡尔曼滤波器[13]]则会阻止系统的复杂化。

利用最近提出的基于变分推断的估计，以及随机梯度变分贝叶斯[14, 20]，隐变量的近似推断变得较为容易。文献[1, 4]将此成果拓展到了时间序列上，而实验表明它们在边界数据概率计算(比如压缩)方面效果提升显著，但在信息完全、状态隐藏的情况下表现平平，这导致无法将其应用在诸如长期采样等方面。然而，在广泛的实际应用中，针对信息完全、状态隐藏的推断要比压缩之类的应用更有价值。同时如果要将隐空间(latent space)应用于下游应用，针对信息完全、状态隐藏的推断也是至关重要的。

针对上述问题，我们提出了一些自己的方法，贡献点总结如下：

(i) 我们首次提出了潜在状态空间模型假设，可以进行可靠的系统识别，并能够对可观测系统进行长期预测；

(ii) 我们为模型提供了具有丰富理论支撑的推断机制；

(iii) 我们提出的模型集成了神经元结构的优点，可以在图像或其它传感器输入的原始数据上进行训练；

(iv) 我们的模型可以用于处理大规模数据，这得益于对基于SGD [3]的参数优化。因此，我们的模型有能力利用系统理论方法去解决下游任务(比如基于控制或模型的强化学习[21])。

2 背景知识及相关工作

2.1 动力系统的概率模型与滤波

我们将一个非线性动态系统看做是由**观察结果**$\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^{n_x}$以及**控制输入或动作**$\mathbf{u}_t \in \mathcal{U} \subset \mathbb{R}^{n_u}$构成的。$\mathcal{X}$中的元素可以是高维的传感器数据(比如可以是原始图像)。请注意，它们有可能表现出复杂的非马尔可夫变换。与此对应的、长度为T 的离散时间序列记为$\mathbf{x}_{1:T} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$ 与$\mathbf{u}_{1:T} = (\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_T)$ 。而我们关心的正是$p(\mathbf{x}_{1:T}|\mathbf{u}_{1:T})$ 这个概率模型。我们假设此动态模型为：

$$p(\mathbf{x}_{1:T}|\mathbf{u}_{1:T}) = \int p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, \mathbf{u}_{1:T})p(\mathbf{z}_{1:T}|\mathbf{u}_{1:T})d\mathbf{z}_{1:T} \tag{1}$$

其中，$\mathbf{z}_t \in \mathcal{Z} \subset \mathbb{R}^{n_z}$，表示在时间上相对应的潜在序列。也就是说，我们假设了一个生成模型，它有着潜在的动态系统，在动态系统中包含一个**发射模型** $p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, \mathbf{u}_{1:T})$ 与一个**状态转移模型** $p(\mathbf{z}_{1:T}|\mathbf{u}_{1:T})$。我们希望能将这两个子模型学习出来，换句话说就是我们希望进行**潜在系统识别**。为了在下游任务中使用这个识别系统，我们需要找到有效的后验推断分布$p(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})$。三种常见的用例是预测、滤波与平滑，这三个任务分别由$\mathbf{x}_{1:t-1}$、$\mathbf{x}_{1:t}$ 以及$\mathbf{x}_{1:T}$ 进行推断。准确的识别和有效的推断通常是互相的任务，因为一个更大范围的生成模型通常会导致更困难的推断，甚至使得无法进行推断。

状态转移模型对于实现良好的长期预测是至关重要的：一个糟糕的转移模型会导致潜在状态出现分歧。因此，我们通过贝叶斯来特别“关照”它。假设在每个时间步上的转移都会不一样，那么我们对变换参数$\beta_{1:T}$ 使用先验分布正则化。

$$(1) = \iint p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, \mathbf{u}_{1:T}) p(\mathbf{z}_{1:T}|\beta_{1:T}, \mathbf{u}_{1:T}) p(\beta_{1:T}) d\beta_{1:T} d\mathbf{z}_{1:T} \tag{2}$$

为了观测状态空间模型，我们假设发射模型与状态变换模型为：

$$p(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}, \mathbf{u}_{1:T}) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{z}_t) \tag{3}$$

$$p(\mathbf{z}_{1:T}|\beta_{1:T}, \mathbf{u}_{1:T}) = \prod_{t=0}^{T-1} p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t, \beta_t) \tag{4}$$

公式3 与4 中，假设了当前状态$\mathbf{z}_t$ 包含当前观察对象$\mathbf{x}_t$ 以及通过当前控制输入$\mathbf{u}_t$ 与变换参数β_t 得出的下一个状态$\mathbf{z}_t$ 中所需要的全部信息。也就是说$\mathbf{z}_t$ 与我们的观测对象不同，这体现出了马尔科夫行为。

线性高斯模型（LGM）是这个假设的典型例子。以LGM 为例，状态转移模型与发射模型都通过高斯噪声偏移公式进行仿射：

$$\mathbf{z}_{t+1} = \mathbf{F}_t \mathbf{z}_t + \mathbf{B}_t \mathbf{u}_t + \mathbf{w}_t \qquad \mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t) \tag{5}$$

$$\mathbf{x}_t = \mathbf{H}_t \mathbf{z}_t + \mathbf{y}_t \qquad \mathbf{y}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t) \tag{6}$$

典型地，假设给定状态转移矩阵$\mathbf{F}_t$ 与控制输入矩阵$\mathbf{B}_t$，则有$\beta_t = \mathbf{w}_t$ 第3.3 章详细描述了我们的方法是怎样将其它变量作为$\beta_t = (\mathbf{F}_t, \mathbf{B}_t, \mathbf{w}_t)$。在LGM 5 与6 的强假设下，我们可以用著名的卡尔曼滤波器对推断进行优化。文献[12]将卡尔曼滤波器推广到非线性动态系统，并成功在多个领域进行了应用。但他们仍旧具有一些缺点：第一，他们的假设在实际应用中受到了各种各样的限制，并且相对独立，这导致最终结果不理想；第二，为了进行后向推断，必须先明确$\mathbf{F}_t$、$\mathbf{B}_t$ 等参数。随后，人们也做了许多努力去优化这种动态系统，如文献[6, 10]利用最大似然算法进行了尝试，文献[23]则利用神经网络进行了尝试。然而这些算法在后验分布很难处理的情况下并不适用，比如在使用图像等后验分布高度非线性的序列时，算法对后验近似为典型平均场的假设过于简单了。而我们提出的新方法将解决这两个问题，并且能够通过随机梯度变分贝叶斯来联合学习进行识别和推断。

2.2 面向时序分布的随机梯度变分贝叶斯

使用随机单元$\mathbf{z}$ 代替一个已知的自编码器的瓶颈层，此时变分自编码器（VAE）[14, 20]会通过图模型以无监督的方式学习$\mathbf{x}$的复杂的边际数据分布：

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \int p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z}$$

在VAE中，$p(\mathbf{x}|\mathbf{z}) \equiv p_\theta(\mathbf{x}|\mathbf{z})$ 是一种利用参数为θ的神经网络的典型参数化。在此框架内，模型通过随机梯度去最大化编辑数据对数似然的下界来进行训练：

$$\ln p(\mathbf{x}) \geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\ln p_\theta(\mathbf{x}|\mathbf{z})] - KL(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) =: \mathcal{L}_{SGVB}(\mathbf{x}, \phi, \theta) \tag{7}$$

可以证明上式等价于对近似后验/识别模型$q_\phi(\mathbf{z}|\mathbf{x})$ 与难以处理的真实数据/后验分布$p(\mathbf{z}|\mathbf{x})$ 间的KL散度进行最小化操作。其中，q_ϕ 为参数为θ 的神经网络参数化。

目前已经有研究将VAE的原理推广到时间序列[1, 4]，但他们都在潜在空间中使用了非线性状态变换，在变换过程中进行了观测，这违反了方程式4。根据经验来说，这种方法对于压缩与重建有很好的效果，但是对于状态空间$\mathcal{Z}$ 不会反映出所有的可用信息，因此使得合理地生成模型，用于长期预测是不可能的。文献[22]已经分析了生成模型的这种现象。

文献[16]对3 和4 的状态空间假设使用深度卡尔曼滤波器（DKF）模型进行了软编码。但是，如第4节中所示，他们的模型无法取得诸如速度（以及一般的时间导数）之类的信息，因此在预测时同样会有类似的问题。

文献[11]则不再局限于动态系统，给出了一般图模型变分推断的算法。与前段描述的方法不同，它没有违背公式4。这种方法的不同之处在于，它将识别模型输出的潜状态与推断潜状态的信息结合了起来。而我们的方法侧重于控制任务相关的动态系统的学习，因此使用神经网络直接推断潜状态，而不推断子路径。

目前对受控动态系统的变分推断的研究也较多。文献[24]E2C）使用VAE学习隐空间的映射，由方程7进行正则化。但是他们仍然没能达到数学上正确的边际数据的下界。并且更重要的是，他们方法中的识别模型需要包含全部信息（包括当前状态w，r，t）的观察结果。因此最大的缺陷就是对数据做了额外的时间上的假设：他们将多个原始样本堆叠在同一个训练样本中，使得潜在的因素（尤其是各种时间导数）都存在于一个样本中。这样就使得任务就变得很简单，因为此时我们学习的是一个基于处理好的数据的静态自编码器，而非原本的时间序列。

我们可以总结出来的是：好的预测能够增强压缩的效果。然而，前面提到的方法在经验上都是对压缩进行了优化，但预测效果却没有提升。我们推测这是由于前面的方法都是试图将潜在的动态模式转变为潜状态，使得其对重建会有利。同时这也有助于学习静态的自编码器，让模型尽可能地从单个观测结果中提取重点，更重要的是，它们不需要为了完美重建单个时间步去了解整个序列。但一旦设定了潜状态，就很难再去调整他们的变换。因此，需要对潜状态进行一些修改，以暂时降低重建效果为代价，让学习算法陷入局部最优，以此来保证不错的重建效果与压缩效果。有趣的是，E2C可以通过扩充数据来绕过这个问题。

这也造就了本文的另一个关键贡献点：我们反过来强制隐空间去拟合转换过程，从而实现状态空间模型假设并获取潜状态的全部信息。

3 深度变分贝叶斯过滤器

3.1 重新定义转换

动态系统学习潜在状态的关键是有效地推断服从状态空间模型假设的隐空间。如果能证明服从，则隐空间必须包含了全部的信息。前面的方法都强调的是良好的重建效果，因此空间仅包含重建所需的一个时间步的信息。为了克服这点，我们建立了在全部时间上变换的梯度路径，从而让变换过程成为了规划隐空间的潜在驱动因素，而不再将转换过程变成识别模型的隐空间。一切的关键就是防止识别模型$q_\phi(\mathbf{z}_{1:T}|\mathbf{x}_{1:T})$ 直接对潜在状态$\mathbf{z}_t$ 进行描绘。

与[14] 的重参数化及其类似，[20] 利用蒙特卡洛估算可微分w,r,t参数，我们对可微w,r,t 的最后的状态及参数进行了转换：

$$\mathbf{z}_{t+1} = f(\mathbf{z}_t, \mathbf{u}_t, \beta_t) \tag{8}$$

给定随机参数β_t，状态转移是确定的（反过来也就是说将β_t边际化，我们依然有着随机转移）。可以得到直接、关键的结论：从z_t重建x_t的误差直接通过时间进行反向传播。

这种重参数化有一些其它的重要意义：识别模型不再对潜状态z_t进行推断，而去推断转换参数β_t。特别地，公式8 很好地定义了梯度$\partial\mathbf{z}_{t+1}/\partial\mathbf{z}_t$，梯度信息可以通过转换进行反向传播。

这个方法与[16]不同，此方法的转换仅发生在其损失函数的KL-散度项中（方程7的变体）。来自生成模型的梯度没有通过转换进行反向传播。

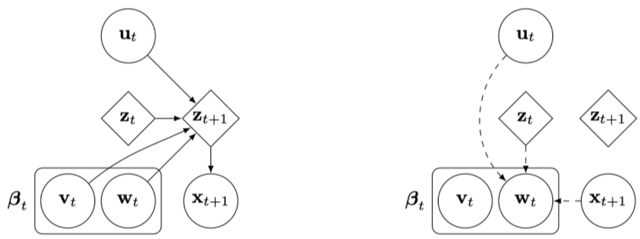
与公式5 一样，随机参数包括校正偏移项$\mathbf{w}_t$，它能够强调识别模型作为滤波器的概念。在理论上来说，算法仍然可以学习$\mathbf{z}_{t+1} = \mathbf{w}_t$ 的转换。然而，引入β_t 也使我们能够通过有意义的先验来对过渡进行正则化，这不仅可以防止识别模型出现过拟合，还可以通过过渡先验在隐空间中进行有意义的集合。如果忽略随着时间的推移发生的变化，会让这些先验会产生巨大的惩罚。因此，可以用构造来克服第2节中概述的问题。

为了应用这样的先验变换，我们将$\beta_t = (\mathbf{w}_t, \mathbf{v}_t)$ 拆开看。$\mathbf{w}_t$ 是一个独立于样本的过程噪声，可以由输入数据推断出来，如公式5 所示。而$\mathbf{v}_t$ 是通用转换参数，它与样本无关（且仅能在训练时由数据推断而出）。这与[9] 提出的权重不确定性相符合。这种解释可以假设识别模型进行自然因子分解：

$$q_\phi(\beta_{1:T}|\mathbf{x}_{1:T}) = q_\phi(\mathbf{w}_{1:T}|\mathbf{x}_{1:T})q_\phi(\mathbf{v}_{1:T}) \tag{9}$$

当使用训练好的模型进行生成采样时（即无输入采样），通用状态转换参数仍然可以从$q_\phi(\mathbf{v}_T 1:T)$ 中提取，而$\mathbf{w}_{1:T}$ 是在无输入数据的情况下从先验中提取的。

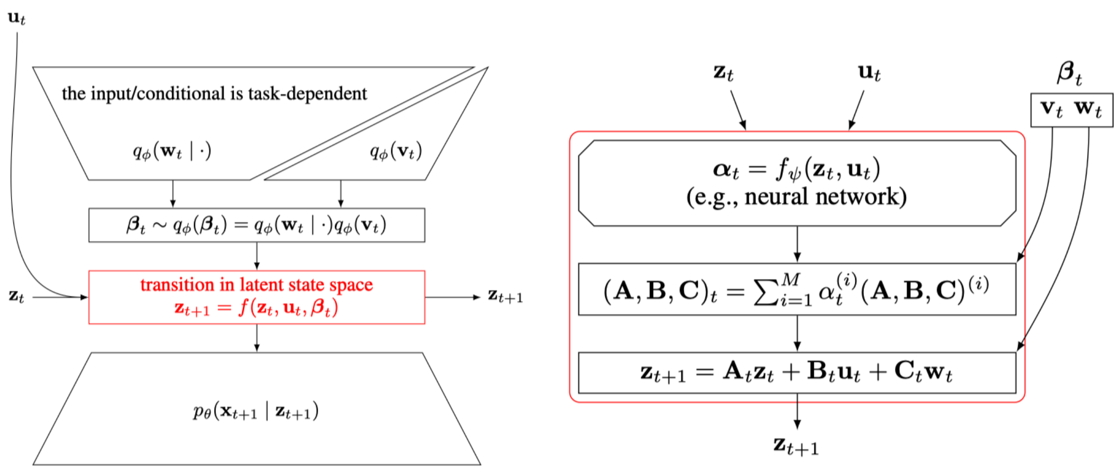
图1展示了底层的图模型和推断过程。图2a展示了我们新的计算架构的一般图示。第3.3节将以局部线性变换参数化为例进行说明。



(a) Forward graphical model.

(b) Inference.

Figure 1: 左图：在状态空间模型假设下进行一次转换的图形模型。更新后的潜在状态$\mathbf{z}_{t+1}$取决于先前的状态$\mathbf{z}_t$、控制输入$\mathbf{u}_t$以及转换参数β_t。$\mathbf{z}_{t+1}$包含用于生成观测值$\mathbf{x}_{t+1}$的全部信息。菱形节点表示对父节点的依赖性。右图：在训练期间（或在过滤时）进行推理。过去的观察结果将间接用于推理，因为$\mathbf{z}_t$包含了有关它们的全部信息。



(a) General scheme for arbitrary transitions.

(b) One particular example of a latent transition: local linearity.

Figure 2: 左图：DVBF的一般架构。通过神经网络之类的识别模型推断随机转移参数β_t。基于采样β_t，可以确定性地计算状态的转换。更新后的潜在状态$\mathbf{z}_{t+1}$用于预测$\mathbf{x}_{t+1}$。3.1节中详细地进行了说明。右图：放大隐空间的转换（左图中的红色框）。以3.3中的局部线性变换为例展示了一种变换。

3.2 下界目标函数

与公式7类似，我们现在得出边际似然$p(\mathbf{x}_{1:T}|\mathbf{u}_{1:T})$ 的下界。在概率因式分解公式2 中分析马尔科夫假设3 与4 后，我们可得到：

$$p(\mathbf{x}_{1:T}|\mathbf{u}_{1:T}) = \iint p(\beta_{1:T}) \prod_{t=1}^{T} p_\theta(\mathbf{x}_t|\mathbf{z}_t) \prod_{t=0}^{T-1} p(\mathbf{z}_{t+1}|\mathbf{z}_t, \mathbf{u}_t, \beta_t) d\beta_{1:T} d\mathbf{z}_{1:T}$$

由于给定β_{t+1}的确定性转换，最后一项狄克拉分布的乘积，并对整体最大分布简化：

$$p(\mathbf{x}_{1:T}|\mathbf{u}_{1:T}) = \int p(\beta_{1:T}) \prod_{t=1}^{T} p_\theta(\mathbf{x}_t|\mathbf{z}_t)|_{\mathbf{z}_t = f(\mathbf{z}_{t-1}, \mathbf{u}_{t-1}, \beta_{t-1})} d\beta_{1:T}$$

$$\left(= \int p(\beta_{1:T}) p_\theta(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}) d\beta_{1:T}\right)$$

上述最后一个公式是用于更简洁地记录公式：公式$p_\theta(\mathbf{x}_{1:T}|\mathbf{z}_{1:T})$ 不独立于$\beta_{1:T}$ 与$\mathbf{u}_{1:T}$ 。下面我们推导目标函数，即数据的下界：

$$
\begin{aligned}
\ln p(\mathbf{x}_{1:T}|\mathbf{u}_{1:T}) &= \ln\int p(\beta_{1:T})p_\theta(\mathbf{x}_{1:T}|\mathbf{z}_{1:T})\frac{q_\phi(\beta_{1:T}|\mathbf{x}_{1:T},\mathbf{u}_{1:T})}{q_\phi(\beta_{1:T}|\mathbf{x}_{1:T},\mathbf{u}_{1:T})}d\beta_{1:T} \\
&\geq \int q_\phi(\beta_{1:T}|\mathbf{x}_{1:T}|\mathbf{u}_{1:T})\ln(p_\theta(\mathbf{x}_{1:T}|\mathbf{z}_{1:T})\frac{p(\beta_{1:T})}{q_\phi(\beta_{1:T}|\mathbf{x}_{1:T}|\mathbf{u}_{1:T})})d\beta_{1:T} \\
&= \mathbb{E}_{q_\phi}[\ln p_\theta(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}) - \ln q_\phi(\beta_{1:T}|\mathbf{x}_{1:T}|\mathbf{u}_{1:T}) + \ln(\beta_{1:T})] \quad (10)\\
&= \mathbb{E}_{q_\phi}[\ln p_\theta(\mathbf{x}_{1:T}|\mathbf{z}_{1:T})] - KL(q_\phi(\beta_{1:T}|\mathbf{x}_{1:T}|\mathbf{u}_{1:T})||\ln(\beta_{1:T})) \quad (11)\\
&=: \mathcal{L}_{DVBF}(\mathbf{x}_{1:T},\theta,\phi,\mathbf{u}_{1:T})
\end{aligned}
$$

我们经过实验发现，公式10 使用退火版整体性能会更好：

$$10 = \mathbb{E}_{q_\phi}[c_i \ln p_\theta(\mathbf{x}_{1:T}|\mathbf{z}_{1:T}) - \ln q_\phi(\beta_{1:T}|\mathbf{x}_{1:T}|\mathbf{u}_{1:T}) + c_i \ln p(\mathbf{w}_{1:T}) + \ln p(\mathbf{v}_{1:T})]$$

此处，$c_i = \max(1, 0.01 + i/T_A)$ 为反温度，会随着梯度更新步数i的增加不断线性增加，直至到达退火迭代次数时T_A 定为1。与这种退火时间规划类似的方法已经有了许多应用[7, 17, 19]，在这些应用中已经体现这种方法在高度非凸情况下的平滑性。此外，在优化期间通过经验贝叶斯等方法估计先进行先验变化$p(\mathbf{v}_{1:T})$。在我们的所有实验中，我们都使用了各向同性高斯先验方法。

3.3 局部线性变换

我们已经推导了出一种学习时间序列的算法，这种算法会特别关注隐空间中的一般转换。本节受到[24]的启发，将以局部线性状态转换为例，展示如何学习特定实例。也就是说，我们将公式8 设为：

$$\mathbf{z}_{t+1} = \mathbf{A_t z_t} + \mathbf{B_t u_t} + \mathbf{C_t w_t}, \qquad \mathbf{t = 1, ..., T}, \tag{12}$$

其中$\mathbf{w}_t$ 为来自识别模型的随机样本，$\mathbf{A_t}$，$\mathbf{B_t}$，$\mathbf{C_t}$ 为维数匹配的矩阵。它们也是$\mathbf{z}_t$及$\mathbf{u}_t$ 的随机函数（因此有着局部线性）。由此，我们可以将$q_\phi(\mathbf{v}_t)$ 记为：

$$\mathbf{v}_t = \{\mathbf{A_t^{(i)}}, \mathbf{B_t^{(i)}}, \mathbf{C_t^{(i)}} | \mathbf{i = 1, ..., M}\},$$

在三个矩阵中，每个矩阵都对应一个与数据无关的全局学习线性系统。这个式子可以应用点估计来进行学习。我们使用一种贝叶斯解法[2]来解决这个问题。我们将$\mathbf{A_t}$，$\mathbf{B_t}$，$\mathbf{C_t}$各自作为状态和控制相关的线性组合：

$$\alpha_\mathbf{t} = \mathbf{f}_\psi(\mathbf{z_t}, \mathbf{u_t}) \in \mathbb{R}^\mathbf{M}$$

$$\mathbf{A_t} = \sum_{\mathbf{i=1}}^{\mathbf{M}} \alpha_\mathbf{t}^{(\mathbf{i})} \mathbf{A_t^{(i)}} \qquad \mathbf{B_t} = \sum_{\mathbf{i=1}}^{\mathbf{M}} \alpha_\mathbf{t}^{(\mathbf{i})} \mathbf{B_t^{(i)}} \qquad \mathbf{C_t} = \sum_{\mathbf{i=1}}^{\mathbf{M}} \alpha_\mathbf{t}^{(\mathbf{i})} \mathbf{C_t^{(i)}}$$

计算结果如图2 b 所示。函数f_ψ 可以看做为一个权重确定为ψ 的神经网络。作为生成式参数θ 的子集，ψ 也是我们模型可训练参数的一部分，权重向量α_t 在三个矩阵间共享。与公式5中的矩阵对应，两个公式中的$\mathbf{A_t}$ 与$\mathbf{F_t}$、$\mathbf{B_t}$ 与$\mathbf{B_t}$、$\mathbf{C_t C_t^T}$ 与$\mathbf{Q_t}$分别一一对应。

我们在实验中使用了状态转移模型的参数化。具体如何参数化取决于用户以及相应的应用。

4 实验及结果

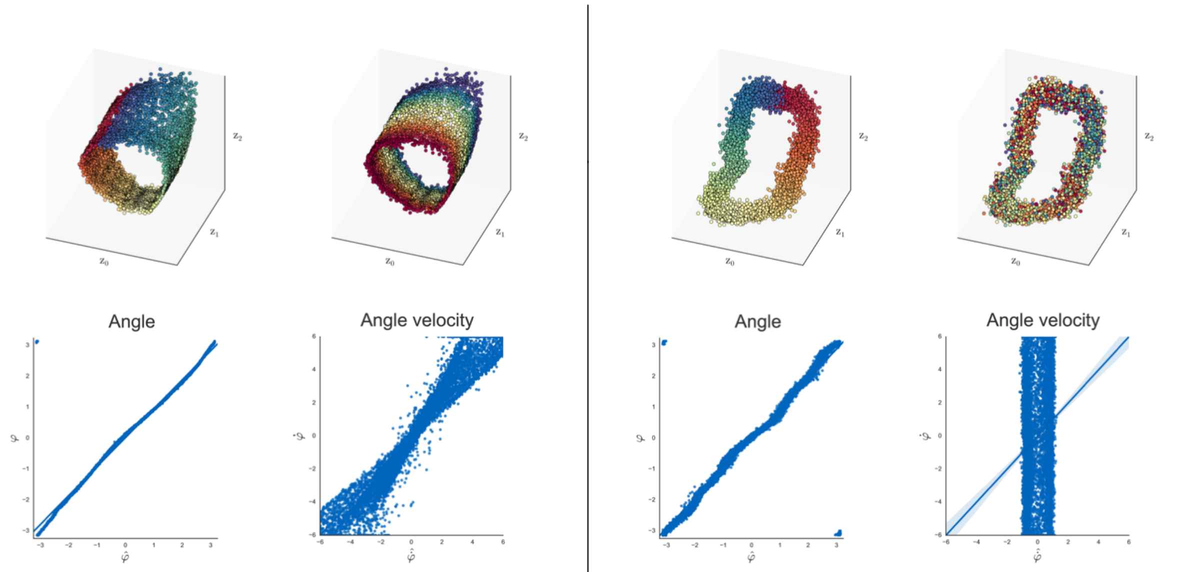
在本节中，我们将验证具有局部线性转换的DVBF（DVBF-LL）（第3.3节）在恢复具有完整信息的隐空间时的效果优于深度卡尔曼滤波器（DKF[16]）。我们主要专注于可以充分正确全面了解潜在动态系统进行模拟的环境。实验的设置将在补充材料中进行描述。我们将DVBF代码公布于

https://brml.org/projects/dvbf。

4.1 动态摆动

为了测试我们的算法在真正的非马尔可夫动态系统中的观测效果，我们模拟了一个由下列微分方程控制的动态扭矩控制摆动系统：

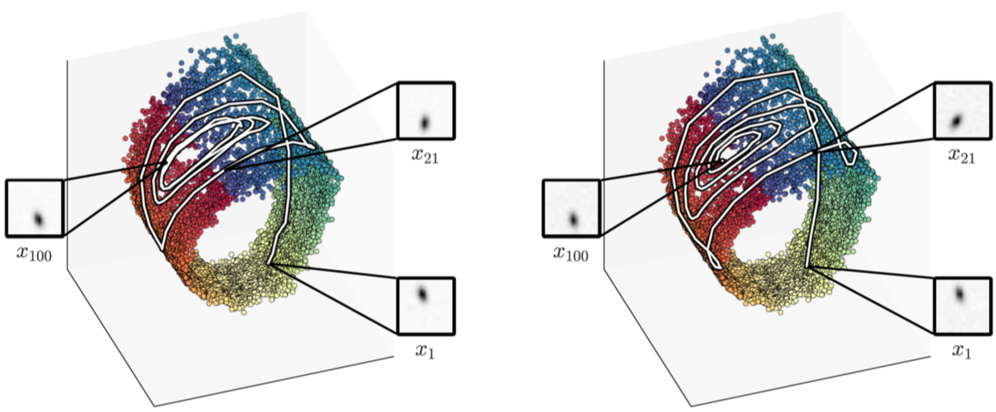
$$ml^2\ddot{\varphi}(t) = -\mu\dot{\varphi}(t) + mgl\sin\varphi(t) + u(t)$$



(a) DVBF-LL

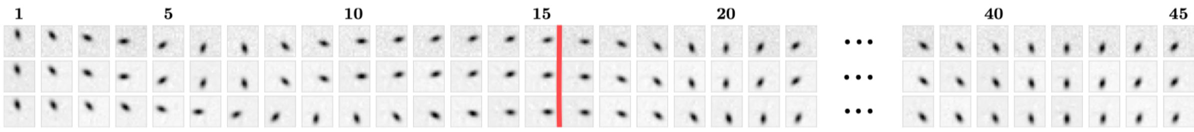
(b) DKF

Figure 3: (a)我们的DVBF-LL模型在摆锤图像序列上的训练结果。上图展示了根据真值着色的隐空间，左图为角度，右图为角速度。下方的图展示了从潜在表示中预测真值的回归结果。隐空间图清楚地表明，用于表示摆锤的完整状态的所有信息都被编码在每个潜在状态中。(b)DKF[16]在同一个摆锤数据集上进行训练的结果。隐空间图显示DKF无法学习出摆锤的速度。因此，它无法完整捕获表示摆状态的所有信息。



(a) Generative latent walk.

(b) Reconstructive latent walk.



(c) Ground truth (top), reconstructions (middle), generative samples (bottom) from identical initial latent state.

Figure 4: (a)生成模型隐空间中的步长。(b)滤波模式下的隐空间步长。(c)识别模型与生成模型中的真值与样本。重建采样过程可以获得观察序列并进行滤波；生成采样只能观测一次序列以创建初始状态，后续所有的样本都是从这个初始状态进行预测的。红色线条表示训练序列的长度。超出红线的样本显示了模型能泛化至比训练序列更长的时间。完整的序列可以在附录中图7找到。

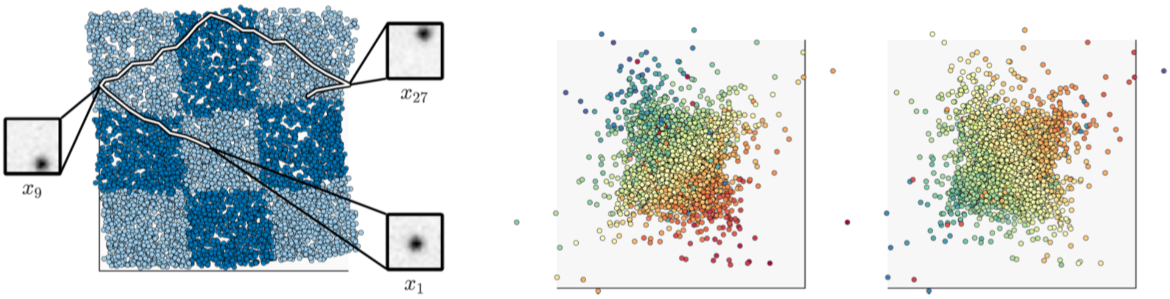
其中$m = l = 1$, $\mu = 0.5$, $g = 9.81$。通过数值积分，将真值φ 转换为$\mathcal{X}$ 中可观测的图像，并对角加速度进行一维控制（与从转动扭矩成比率）。角度和角速度可以充分描述系统的状态。

Table 1: 摆锤实验的结果。包含了所有潜在状态的因变量的OLS 回归结果。

		DVBF-LL		DKF	
		Log-likehood.	R^2	*Log-likehood.*	R^2
Dependent	$\sin(\varphi)$	3900.8	0.961	1737.6	0.929
ground truth	$\cos(\varphi)$	7231.1	0.982	6614.2	0.979
variable	$\dot{\varphi}$	-11139	0.916	-20289	0.035

Table 2: 摆动实验中目标函数在测试集上的均值。

	下界=重建错误-KL散度		
DVBF-LL	798.56	802.06	3.50
DKFh	784.70	788.58	3.88



(a) Latent walk of bouncing ball.

(b) Latent space velocities.

Figure 5: (a)为4维弹球隐空间中的两维。真值x 与y 坐标按照3x3标准大小进行着色。这个棋盘状的形态正式从嵌入中提取出来的。(b) 其它两个隐维度。同样的潜在样本，按照球在x 与y 轴上的速度进行着色（分别与左图、右图对应）。平滑、正交的着色结果表明真值存在于隐维度中。

图3 展示了分别由DVBF-LL与DKF学习的、输入相同数据的隐空间，上方的图为真值着色。需要注意的是，展示的是隐样本，而非后验分布。状态空间模型可以使用三个隐维度。如图3a 所示，DVBF-LL学习到了二维流形的嵌入（embedding），即角度在极坐标中的编码（从而避免了角度取2π 模的不连续性）。下面的图展示了一般最小二乘（OLS）回归以演示性能。观察可以发现，DVBF-LL的潜状态与真值角度和角速度之间高度相关。与此相反，图3b验证了我们的预测，DKF同样能够学习角度，但几乎没有提取出角速度的信息。

表1中显示的OLS回归结果证实了这一观察结果。在预测$\sin(\varphi)$ 与$\cos(\varphi)$ （即真值角φ的极坐标）时，DVBF-LL和DKF效果几乎同样好，DVBF-LL稍微优于DKF。但在预测真值角速度$\dot{\varphi}$时，DVBF-LL表现出了优秀的性能。相反，DKF几乎没有得到任何信息，最终得到了$R^2 = 0.035$这样的低分。

图4表明，真值与潜在状态间的强关系对于生成抽样是很有帮助的。所有图都展示了从完全相同的潜状态开始并且未被扰动的摆锤的100个时间步长的状态。上左方的图展示了单纯生成的隐空间步，上右方的图展示了过滤观测校正后的隐空间步。我们可以看到两者都类似于吸引子的轨迹。生成模型在接近吸引子时更容易产生噪音。

下方的图展示了相应观测（第一行）、重建（第二行）和生成样本（未经观察校正）的前45个步。有趣的是，即使序列比训练序列（红线表示序列尾部）长很多，DVBF也能很好地进行预测。

表2展示了边际数据似然的下界（DVBF-LL的结果与公式12对应）。我们看到DVBF-LL在压缩方面优于DKF，但边界非常小，没有像之前工作中那样[22]那样反映出更好的生成采样。

4.2 弹球

弹球实验就是在平面框的边界内滚动球。此系统具有二维的控制输入，直接加到球的速度上。由于球撞上墙时会反弹，因此真正的动状态很大程度上取决于球的当前位置和速度。系统的状态是四维的，位置和速度各有两个维度。

因此，我们使用具有四个隐维度的DVBF-LL。图5展示了DVBF再次在隐空间捕获整个系统动态的效果。如同棋盘一样的结果是一个非常了不起的结果：球的真值位置位于一个2维正方形内，即始终在边框内。为了在学习潜状态时推测真值会如何出现，我们观察到了隐空间在边框出现了翘曲。为此，我们将真值单元正方形离散化为3x3的格子，并对应上色。我们发现DVBF学习出了从256个像素中提取2维的位置，并将它们在隐空间的两个维度上对齐，刚好与物理系统完全对应。此算法可以进行像素到2维空间的推断，人类观察者在看到图像时会自然而然地得出这个推论。

Figure 6: 双弹球实验中，从潜在状态初始化开始后的真值（最上方）、重建（中间）、生成样本（下方）。红线代表训练序列的长度。

4.3 双弹球

设定一个更复杂的环境，在边框中有两个球。我们使用10维隐空间来完全捕捉球的位置和速度信息。重建和生成样本如图6所示。与摆锤实验一样，我们得到的生成模型能够对超出训练数据序列长度的数据进行稳定预测。

5 总结

我们提出了深度变分贝叶斯过滤器(DVBF)，这种新方法可以从非马尔科夫原始数据中学习出状态空间模型。DVBF 能进行潜在动态系统识别，并且能够解决非常困难的推断问题。由于DVBF 使用了随机梯度变分贝叶斯，因此自然而然地可应用于大规模数据集。在一系列基于视觉的实验中，我们证明了此模型可以通过识别出潜藏的物理量恢复潜在状态。生成模型显示出了长期预测的稳定性，且远远超出了训练时使用的序列长度。

References

[1] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

[2] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015.

[3] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[4] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pages 2980–2988, 2015.

[5] Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*, pages 465–472, 2011.

[6] Zoubin Ghahramani and Geoffrey E Hinton. Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Totronto, Dept. of Computer Science, 1996.

[7] Zoubin Ghahramani and Geoffrey E Hinton. Variational learning for switching state-space models. *Neural computation*, 12(4):831–864, 2000.

[8] Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[9] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13. ACM, 1993.

[10] Antti Honkela, Tapani Raiko, Mikael Kuusela, Matti Tornio, and Juha Karhunen. Approximate riemannian conjugate gradient learning for fixed-form variational bayes. *Journal of Machine Learning Research*, 11(Nov):3235–3268, 2010.

[11] Matthew Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in neural information processing systems*, pages 2946–2954, 2016.

[12] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–194. International Society for Optics and Photonics, 1997.

[13] Rudolph E Kalman and Richard S Bucy. New results in linear filtering and prediction theory. *Journal of basic engineering*, 83(1):95–108, 1961.

[14] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[15] Jonathan Ko and Dieter Fox. Learning gp-bayesfilters via gaussian process latent variable models. *Autonomous Robots*, 30(1):3–23, 2011.

[16] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

[17] Stephan Mandt, James McInerney, Farhan Abrol, Rajesh Ranganath, and David Blei. Variational tempering. In *Artificial Intelligence and Statistics*, pages 704–712, 2016.

[18] Kevin McGoff, Sayan Mukherjee, Natesh Pillai, et al. Statistical inference for dynamical systems: A review. *Statistics Surveys*, 9:209–252, 2015.

[19] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

[20] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[21] Ilya Sutskever and Geoffrey Hinton. Learning multilevel distributed representations for high-dimensional sequences. In *Artificial Intelligence and Statistics*, pages 548–555, 2007.

[22] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

[23] Harri Valpola and Juha Karhunen. An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural computation*, 14(11):2647–2692, 2002.

[24] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in neural information processing systems*, pages 2746–2754, 2015.