迈向表示无关概率编程

OLE FENSKE*、MAXIMILIAN POPKO*、SEBASTIAN BADER 和 THOMAS KIRSTE，德国视觉与分析计算研究所

arXiv:2512.23740v1 [cs.PL] 25 Dec 2025

当前的概率编程语言和工具将模型表示与特定推理算法紧密耦合，阻碍了人们尝试新型表示或混合离散-连续模型。我们提出了一种包含五项基本操作的因子抽象，可作为通用接口，无论因子的底层表示如何，都能对其进行操作。这使表示无关的概率编程成为可能，用户可以在单一统一框架中自由混用不同表示（例如离散表、高斯分布、基于样本的方法），从而能够对当前工具包无法充分表达的复杂混合模型进行实际推断。

CCS概念：• **计算方法** → 并发编程语言。

加法　　关键词与短语：因子图、概率编程　　ming

1 引言

概率编程语言（PPLs）和工具包（PPTs）使从业者能够表达复杂的统计模型并执行贝叶斯推断，而无需手动实现推理算法。然而，能够成功定义的模型范围取决于表示分布的机制。值得注意的是，有几种概率编程语言（例如 PyMC[Salvatier 等人 2016], Stan [Carpenter 等人 2017], Pyro [Bingham 等人 2019]）将模型表示与特定的推理算法紧密耦合（例如，Stan 的 NUTS采样器）。PPTs Factorie[McCallum 等人 2009], RXInfer [Bagaev 等人 2023],Infernet [Minka 等人 2018] 则基于“因子”这一概念提供了范围更广的表示，但没有提供定义新表示的标准方法。

然而，高效的概率推理计算通常需要高度专业化的表示。例如，结构化概率空间 [Choi 等人 2015] ——散集上的分布——很难使用基于样本、（稀疏）数组或参数化机制的“标准”表示来表示；相反，它们依赖于概率句子决策图 [Kisa 等人 2014]。因此，对于一个“通用”的概率推理库而言，能够添加此类专用表示将是理想的。此外，人们希望库API能够提供一组操作，以便以独立于所选具体表示的方式来表述所有概率计算。

因子概念 [Koller 和 Friedman 2009], 因子概念推广了分布函数的概念，原则上是一种强大的抽象工具。结合一套标准的因子操作，它能够将模型语法（概率结构）与语义（计算实现）分离。这使得滤波或平滑等推理算法可以表示为因子表达式不受底层表示的影响。因此，我们主张将“因子”作为基本抽象，并在工具包提供的因子级 API 中移除对因子具体表示的任何假设。

本文概述了此类工具因子级 API 的基本结构，并通过一个直观示例说明如何运用这一形式体系。

2 因子与因子运算

因子 f_{XY} 定义在随机变量 X 和 Y 之上，是一个抽象的数学函数 $f\colon\ X \times Y \to \mathbb{R}$ 它为每一种配置赋予一个实数$(x,\ y)$。因子推广了许多熟悉的概念：离散变量的概率表、连续变量的高斯分布、离散与连续混合分布（条件高斯分布）[Lauritzen 1992]以及未归一化势函数都是因子。至关重要的是，这一抽象的数学定义有意与因子在计算机中的表示方式相分离。

五种基本运算构成了用于操作因子的 API，无论因子采用何种表示方式：

表 1. 因子运算。

运算	表达式	用法
乘法	$h_{XYZ} = f_{XY} \otimes g_{YZ}$	连接
求和消元	$g_Y = \oint_X f_{XY}$	边缘化
约简	$g_Y = f_{XY}^{(X=x)}$	条件化
除法	$h_{XYZ} = f_{XY} \oslash g_{YZ}$	平滑
加法	$h_X = f_X \oplus g_X$	混合

前三种操作是传统的标准因子操作，构成了因子图中消息传递（例如和积算法）的代数核心。乘法用于组合消息，求和消元用于对变量进行边缘化，而约简则用于纳入证据 [Kschischang 等人 2001]。除法是概率推断中的标准操作（例如，平滑密度[Koller 和 Friedman 2009]），但概率编程工具并不总是提供该操作（其结果通常无法归一化）。加法——通常仅在概率编程工具内部隐式使用——对于计算分布的混合是必要的；具体而言，在构建层次因子表示时，它也是实现求和消元操作所必需的。

本文着重识别计算所需的核心运算集合：这类计算以分布为输入，并以分布为输出；这构成了概率论核心。还有许多有用的运算以分布为输入，却会产生其他数学对象，例如最大后验估计、互信息或熵；这些运算自然属于更全面的“标准库”。此类派生运算的设计与形式化不在本文讨论范围之内。这种划分体现了语义上的区别：

*两位作者对本研究的贡献相同。

作者联系方式：Ole Fenske，ole.fenske@uni-rostock.de；Maximilian Popko，maximilian.popko@uni-rostock.de；Sebastian Bader，sebastian.bader@uni-rostock.de；Thomas Kirste，thomas.kirste@uni-rostock.de，德国罗斯托克视觉与分析计算研究所。

核心运算在分布上封闭，而派生运算则计算分布的泛函或优化器。

3 表示无关性

因子可以采用多种表示方式：离散因子可以是数组、哈希表或句子决策图；连续因子可以是参数化的（例如高斯均值/方差），也可以是非参数化的（样本）。语法与语义的分离确保了可扩展性：只需为因子算子实现特定于表示方式的方法，即可添加新的领域特定表示，而无需更改核心框架。

随后，便可以在上述因子运算的层面实现推理算法（在 API 层面将其定义为泛型函数）。变量消除 [Koller 和 Friedman2009] 以及置信传播 [Pearl 1988] 例如，自然可以分解为一系列此类因子运算，并且专为因子图上的概率推断而设计。例如，状态空间模型中的滤波涉及以下因子表达式：

(1) 预测步骤: $p_{X_{t+1}|y_{1:t}} = \oint_{x_t} p_{X_{t+1}|X_t} \otimes p_{X_t|y_{1:t}}$**(2) 校正步骤** （基于新的观测 y_{t+1}）：

$$\phi_{X_{t+1},y_{1:t+1}} = p_{X_{t+1}|y_{1:t}} \otimes p_{Y_{t+1}|X_{t+1}}^{(Y_{t+1}=y_{t+1})}$$

$$p_{y_{t+1}|y_{1:t}} = \oint_{x_{t+1}} \phi_{X_{t+1},y_{1:t+1}}$$

$$p_{X_{t+1}|y_{1:t+1}} = \phi_{X_{t+1},y_{1:t+1}} \oslash p_{y_{t+1}|y_{1:t}}$$

下标中的大写字母表示因子的变量，小写字母则只是因子名称的一部分。注意，$p_{y_{t+1}|y_{1:\ t}}$是一个变量列表为空的因子：一个简单的标量。

在这里，每种运算都会分派到特定于表示的实现。这既保持了所定义的因子代数与表示代数之间的同态，也使同一模型中能够混合使用不同表示（例如，将离散因子表示为有限映射，将连续因子表示为高斯分布或样本，或将混合因子表示为嵌套表示），同时还便于无缝扩展。

4 利用表示无关性——一个简单示例

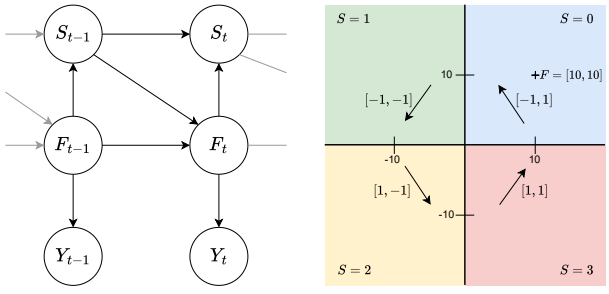

图 1.（左）模型依赖关系的图示；（右）象限及其线性转移模型示意图。

举一个简单的例子，考虑一个划分为四个象限的二维世界，每个象限都有一个主要运动方向，如图所1示。在这个世界中运动的物体会按照其所在象限的方向移动

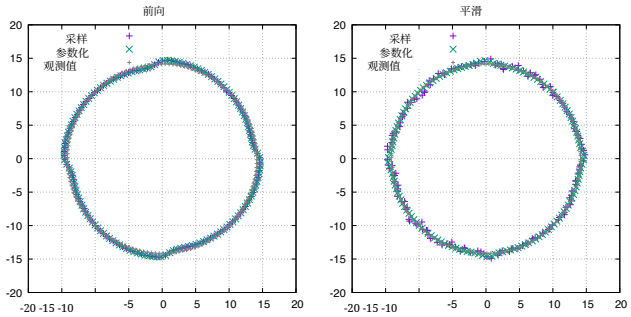

图 2. 基于样本的表示与参数化表示的比较。给定圆形观测值时，$F_{1:T}$状态轨迹的边缘均值。

（并伴有一些高斯噪声）。一旦物体越过象限边界，就会根据其进入的象限改变运动方向。这是一个简单的混合动态模型，其中连续变量 F_t （位置，范围为 $\mathbb{R}^2$）与离散状态 S_t（范围为 {0, 1,2,3}，即二维平面中的四个象限之一）相互作用。图1展示了象限模型的因子化结构，刻画了连续（亚符号）状态 F_t 与离散（符号）状态 S_t随时间的相互作用。连续动力学 $p_{F_t|F_{t-1},S_{t-1}}$描述区域内的平滑运动，而离散动力学 $p_{S_t|S_{t-1},F_t}$则刻画进入下一个象限这一事件。观测因子 $p_{Y_t|F_t}$将潜在过程与测量数据联系起来。这些因子共同定义了模型完整的概率结构。连续演化与离散转移的耦合构成了一个简单的混合系统，展示了因子化表示如何将符号推理与连续状态估计统一起来：描述概率计算的因子表达式不受所涉及的随机变量类型及所选表示方式影响。

主要的计算挑战出现在条件高斯转移因子 $p_{F_t|F_{t-1},\ S_{t-1}}$与不连续链接因子 $p_{S_t|S_{t-1},\ F_t}$的乘法(⊗)运算过程中。这种模型无法用简单的参数化方式表示。模型开发者可能会想要比较不同的近似表示方案，例如，将基于样本的表示与使用截断高斯分布和矩匹配的参数化近似进行比较。

基于采样的表示通过蒙特卡洛估计或基于粒子的更新来近似这些运算。参数化表示以闭式形式表示因子（例如截断高斯分布和矩匹配），并以解析方式执行所有运算。两者使用相同的一组通用因子运算，但定义计算实现的方法不同。

这种分离确保概率模型可以保持不变，同时能够在单一统一框架内灵活探索不同的因子表示和推断方案。图 2 展示了参数化表示和采样表示的滤波与平滑轨迹，这只需替换底层因子定义即可实现。

5 结论

如果使用与表示无关的通用操作来定义因子表达式，就可以实现：(1) 在同一模型中组合 不同的表示，(2) 交换 不同的表示以进行实验，以及 (3) 添加新的专用表示——而无需修改给定模型。

我们的观点并不是现有的概率编程语言/概率编程工具缺少用于概率计算的面向对象 API，而是表示无关性似乎迄今并未成为模型构建层面的一项明确设计目标——我们认为，未来的发展应当采用这一目标。

通过将因子和因子运算视为一等抽象，我们实现了表示无关的概率编程，使推理算法能够统一处理异构表示。用户可以自由尝试不同的表示策略，并在统一框架内扩展系统，加入领域特定表示。这种方法使得对复杂混合模型进行实际推断成为可能，而当前的工具包无法充分表达或高效求解这类模型。

展望未来，一个值得探讨的问题是：对分布进行的计算与基于采样的概率程序有何关联。执行概率程序会得到从某个（通常是联合）分布中抽取的实现值。如果将这些实现值视为程序的动态语义，那么产生这些实现值的分布便可视为程序的静态语义，亦即其语义类型。按照这种解释，本文所确定的运算构成了计算概率程序类型的最小核心。近期研究，例如 [Faggian 等人 2024] 探讨了如何使用类型化 λ 演算构建概率程序类型；我们的方法通过为计算和表示此类类型提供灵活的、以分布为中心的基础，与这条研究路线相衔接。

参考文献

Dmitry Bagaev、Albert Podusenko 和 Bert de Vries。2023 年。RxInfer：用于响应式实时贝叶斯推断的 Julia 软件包。《开放源码软件期刊》8，84（2023），5161。doi:10.21105/joss.05161 Eli Bingham、Jonathan P. Chen、Martin Jankowiak、Fritz Obermeyer、Neeraj Pradhan、Theofanis Karaletsos、Rohit Singh、Paul Szerlip、Paul Horsfall 和 Noah D. Goodman。2019 年。Pyro：深度通用概率编程。20，1（2019），973–978。Bob Carpenter、Andrew Gelman、Matthew D. Hoffman、Daniel Lee、Ben Goodrich、Michael Betancourt、Marcus Brubaker、Jiqiang Guo、Peter Li 和 Allen Riddell。2017 年。*Stan*：一种概率编程语言。76，1（2017）。doi:10.18637/jss.v076.i01 Arthur Choi、Guy Van den Broeck 和 Adnan Darwiche。2015 年。结构化概率空间的高效学习：偏好分布学习案例研究。https://www.semanticscholar.org/paper/Tractable-Learning-for-Structured-Probability-A-in-Choi-Broeck/72bb6887e437942f829822cd9090f63672dd3441 Claudia Faggian、Daniele Pautasso 和 Gabriele Vanoni。2024 年。高阶贝叶斯网络：精确方法。《*ACM* 编程语言汇刊》8，POPL（2024 年 1 月），84:2514–84:2546。doi:10.1145/3632926Doga Kisa、Guy Van den Broeck、Arthur Choi 和 Adnan Darwiche。2014 年。概率句子决策图。载于第十四届知识表示与推理原理国际会议论文集（奥地利维也纳，2014-07-20）（*KR'14*）。AAAI 出版社，558–567。Daphne Koller 和 Nir Friedman。2009 年。《概率图模型：原理与技术——自适应计算与机器学习》。麻省理工学院出版社。F.R. Kschischang、B.J. Frey 和 H.-A. Loeliger。2001 年。因子图与和积算法。《*IEEE* 信息论汇刊》47，2（2001），498–519。doi:10.1109/18.910572Steffen L. Lauritzen。1992 年。《混合图关联模型中的概率、均值与方差传播》。《美国统计学会杂志》87，420（1992），1098–1108。arXiv:https://www.tandfonline.com/doi/pdf/10.1080/01621459.1992.10476265 doi:10.1080/01621459.1992.10476265

Andrew McCallum、Karl Schultz 和 Sameer Singh。2009 年。FACTORIE：通过命令式定义因子图实现概率编程。载于神经信息处理系统（*NIPS*）。T. Minka、J.M. Winn、J.P. Guiver、Y. Zaykov、D. Fabian 和 J. Bronskill。2018 年。/Infer.NET 0.3。微软剑桥研究院。http://dotnet.github.io/infer。朱迪亚·珀尔。1988 年。《智能系统中的概率推理：可信推断网络》。摩根·考夫曼出版社，旧金山，加利福尼亚州，美国。John Salvatier、Thomas V. Wiecki 和 Christopher Fonnesbeck。2016 年。使用 PyMC3 在 Python 中进行概率编程。2（2016），e55。doi:10.7717/peerj-cs.55 出版方：PeerJ Inc.。