高斯过程门控分层专家混合模型

刘宇豪，*M ember*，*IEEE*，Marzieh Ajirak，*IEEE* 会员，以及 Petar M. Djuri´c，*IEEE* 会士*E*

摘要——本文提出一种新颖的高斯过程门控分层专家混合模型（GPHME）。与其他采用输入线性门控模型的专家混合模型不同，我们的模型使用由高斯过程（GP）构建的门控函数。这些过程基于随机特征，而随机特征是输入的非线性函数。此外，我们模型中的专家也由高斯过程构建。GPHME 的优化通过变分推断完成。所提出的 GPHME 具有多项优势。与在输入空间中划分数据的基于树的层次专家混合基准相比，它们表现更优，而且能够以较低的复杂度取得良好性能。另一项优势是，它们能提升深层高斯过程的可解释性，更广泛地说，也能提升深层贝叶斯神经网络的可解释性。即使数据集规模相当有限，我们的 GPHME 在大规模数据集上也展现出卓越的性能。

索引词—高斯过程、专家混合模型、软决策树和随机特征。

arXiv:2302.04947v2 [cs.LG] 3 Mar 2024

1 引言

本文提出了一种利用高斯过程（GP）构建分层专家混合模型（HME）的新方法。分层专家混合模型已在众多应用中发挥重要作用，涵盖回归、分类以及融合任务，涉及医疗保健、金融和模式识别等多个领域 [1]。这些模型可被视为条件混合模型，其中目标变量的分布表示为多个专家分布的组合。专家分布及其相应的混合系数均以输入变量为条件。传统上，模型参数通过最大化似然来估计，这种做法往往会导致严重的过拟合。为应对这一挑战，文献 [2]提出了一种基于变分推断的全面贝叶斯模型处理方法。

本文的工作与文献 [3]更为契合。该文作者为分层专家混合模型提出了一种端到端可微的摊销变分推断算法。该算法采用循环神经网络，近似树节点路由决策的后验分布。这不仅提升了层次专家混合框架的灵活性和效率，也彰显了该领域方法的持续演进。

分层专家混合与具有固定树结构的软决策树在结构上相似。我们回顾一下，决策树是一种由内部节点和终端叶节点组成的层次结构 [4], [5]。典型的决策树由表示属性测试的内部节点组成。根据测试结果，树将样本分配给某个子节点。另一方面，叶节点存储分类任务的标签，或存储回归任务所用的常数。因此，样本会沿着从根节点到某个叶节点的一条路径前进。

决策树节点具有基于特征比较的特性。单变量节点使用单个特征，并将其与阈值 [5] 进行比较。多变量节点则在输入空间中定义线性判别函数进行比较 [6], [7], ，并能够将该判别函数推广为非线性形式 [8]。决策树可以在各自的节点上实现各种统计模型选择程序 [9]。

与子节点由确定性规则选出的硬决策树不同，软概率决策树会以特定概率选择子节点 [10]。这意味着要遍历所有通往叶节点的可能路径，最终决策则由所有叶节点按照不同概率共同作出。

在可解释的基于特征的分类中，软决策树的可解释性也适用于高斯软决策树 [11]。在 [12], 中，研究人员使用深度神经网络（DNN）训练软决策树，使其模仿神经网络（NN）所发现的输入输出函数。软决策树为深度神经网络提供可解释性，并使单个隐藏层的表征具备可解释性。这种方法类似于采用参数化概率树聚合的分层专家混合 [1], ，其中专家对应叶节点，系数则由门控节点得出。

人们也研究了高斯过程扩展与分层专家混合模型的结合， [13], [14], 并探讨了 [15], 分层高斯过程潜变量模型。我们观察到，基于高斯过程的分层专家混合模型主要依赖函数空间，而非特征空间。

多篇论文 [16], [17], [18], [19]探讨了高斯过程与神经网络之间的联系。鉴于高斯过程与具有无限多个隐单元的单层神经网络之间的关系， [20]，高斯过程通过隐式地使用无限表征，缓解了指定隐藏层中单元数量这一难题。这在特征空间的背景下尤其有益，因为高斯过程模型有助于构建能够量化不确定性的贝叶斯神经网络（BNN）。

稀疏高斯过程能够将标准高斯过程的计算复杂度从立方级降至平方级，因此尤为突出。降低高斯过程复杂度的主要方法有三种：（1）基于诱导点的方法 [21],（2）使用随机傅里叶特征进行有限基展开

（RFF） [22], 以及（3）协方差算子的希尔伯特空间近似（希尔伯特空间高斯过程） [23]。与基于诱导点的方法不同，基于随机傅里叶特征的高斯过程和希尔伯特空间高斯过程无需进行矩阵分解，只需计算矩阵乘积，从而提升计算速度。这两种方法会将非线性输入空间转换为线性核特征空间，以下称为特征空间，以区别于原始输入空间。

本文采用基于随机傅里叶特征的高斯过程。将随机傅里叶特征展开应用于高斯过程的各种核函数，会得到神经网络中不同的激活函数。例如，三角函数与径向基函数（RBF）核相关，而 ARC-COSINE 核则对应修正线性单元（ReLU）函数 [24]。在使用平方指数核时，针对随机傅里叶特征后验分布的变分学习方法由 [25] 提出。

本领域近期与我们工作相关的研究包括贝叶斯树状高斯过程模型 [26], 、高斯过程贝叶斯加性回归树（GP-BART） [27], 和高斯过程树[28]。贝叶斯树状高斯过程模型通过树状划分结合平稳高斯过程，从而实现非平稳建模。 [27] 提出的 GP-BART 方法引入了一种基于高斯过程的贝叶斯加性回归树方法。该方法的特点是在加性回归树框架中使用高斯过程。具体而言，在标准 BART 中，观测值之间没有协方差结构，这可能导致模型性能不佳。GP-BART 模型通过假设每个叶节点预测都服从高斯过程先验，扩展 BART 以弥补这一缺陷。 [28], 中，作者提出了一种专为小样本增量学习场景设计的高斯过程树型分类器。该模型应对了利用有限带标签样本进行学习的挑战，在数据稀缺的场景中表现出色。该领域近期的其他研究还包括 [28]。

上述模型仅在输入空间中划分数据，形式上要么是分层专家混合模型，要么是软决策树。换言之，门控模型或内部节点是输入的非线性函数。本文提出了由高斯过程门控的混合专家模型（GPHME），其结构为固定的完全二叉树，门控模型和专家模型均基于高斯过程的随机特征展开。这使高斯过程能够在线性特征空间中进行决策，并大幅降低树的复杂度。我们通过变分推断并利用重参数化技巧来优化 GPHME。实践结果表明，即使对于拥有数百万个样本的大规模数据集，树的最优高度也不超过四。树结构规模较小，是因为输入空间经过了非线性变换，并形成了斜决策机制。此外，GPHME 的表现优于那些对输入数据进行划分的基于树的基准方法。

我们的模型也可用于提高深层高斯过程（DGP）的可解释性。DGP 是基于高斯过程映射堆叠而成的深度信念网络，其中每个隐藏层都由一个多元高斯过程构成。实际应用中，与深度神经网络相比，DGP 所需的深度和宽度更小。然而，DGP 面临着隐藏层难以解释的问题。与通过软决策树 [12], 提炼深度神经网络的可解释性类似，本文提供了一些方法，帮助深入了解 DGP 如何做出决策。

总而言之，本文提出了 GPHME 模型，这是一种使用基于随机特征的高斯过程的新型混合专家模型。我们还展示了 GPHME 相较于现有最先进方法的优越性。此外，所提出的方法复杂度更低，能够直接应用于大规模问题。借助我们的方法，相较于树，我们能够量化高斯过程概率分布的不确定性，为决策者提供有价值的见解。此外，我们还提供了一种解释 DGP 行为的方法，并对位于叶节点上的模型给出了解释。我们的贡献拓展了当前对分层模型的理解，为未来的研究和应用提供了一个富有前景的方向。

2 背景

高斯过程是一种随机过程（按时间或空间索引的一组随机变量），其中每个有限的随机变量集合都服从多元正态分布。更具体地说，对于有限输入集合$\mathbf{X} \in \mathbb{R}^{N\times D_x}$，其中 N 是输入的数量， D_x 是输入的维度，相应的输出$\mathbf{y} \in \mathbb{R}^N$服从高斯过程 f，其中 $f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}) \kappa(\mathbf{x}, \mathbf{x}'))$，其中$\mu(\mathbf{x})$是该过程的均值， $\kappa(\cdot, \cdot)$ 是核函数或协方差函数。一种常用的核函数是径向基函数（RBF），其定义为

$$\kappa(\mathbf{x}, \mathbf{x}') = \sigma_\lambda^2 \overline{\kappa}(\mathbf{x}, \mathbf{x}') = \sigma_\lambda^2 \exp\left[-\frac{1}{2}\sum_{d=1}^{D_x} \frac{(x_d - x_d')^2}{\lambda_d^2}\right], \quad (1)$$

其中， σ_λ^2 是核方差， x_d 是$\mathbf{x}$的第 d个元素， λ_d 是对应于 x_d的核长度尺度，而κ 是范数为 $||\kappa(\cdot, \cdot)|| \leq 1$的标准化核函数。引入参数 λ_d 是为了对相应的输入进行自动相关性确定（ARD）。

另一种常用的核函数是次数为 n的 ARC-COSINE 核。

$$\widetilde{\kappa}(\mathbf{x}, \mathbf{x}') = \frac{||\mathbf{x}|| \times ||\mathbf{x}'||}{\pi} J_n(\alpha), \quad (2)$$

其中

$$J_n(\alpha) = (-1)^n (\sin\alpha)^{2n+1} \left(\frac{1}{\sin\alpha}\frac{\partial}{\partial\alpha}\right)^n \left(\frac{\pi - \alpha}{\sin\alpha}\right), \quad (3)$$

and

$$\alpha = \cos^{-1}\left(\frac{\mathbf{x}^\top \mathbf{x}'}{||\mathbf{x}||||\mathbf{x}'||}\right). \quad (4)$$

这种核函数之所以称为弧余弦核，是因为它取决于角度 α 和反余弦函数。本文使用 $n = 1$，因为它能生成带有固定乘数的修正线性单元（ReLU）随机特征，并且

$$J_1(\alpha) = \sin\alpha + (\pi - \alpha)\cos\alpha. \quad (5)$$

这个核函数也可以通过将 x_d 除以λ_d来实现 ARD。

2.1 高斯过程的随机特征展开

Bochner 定理指出，如果 $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \kappa(\mathbf{x}_i - \mathbf{x}_j)$ 是连续的、平移不变且归一化的协方差函数，则它可以改写为非负测度 $p(\boldsymbol{\omega})$ [22] 的傅里叶变换。令 $\boldsymbol{\omega}$ 表示一个谱频率向量，$i = \sqrt{-1}$，以及 $\boldsymbol{\Delta} = \mathbf{x}_i - \mathbf{x}_j$。那么我们可以写成

$$\overline{\kappa}(\boldsymbol{\Delta}) = \int p(\boldsymbol{\omega}) \exp\left(i \boldsymbol{\Delta}^\top \boldsymbol{\omega}\right) d\boldsymbol{\omega}, \tag{6}$$

其中 $p(\boldsymbol{\omega})$ 是 $\kappa(\mathbf{x}_i, \mathbf{x}_j)$ 的傅里叶变换。例如，如果核函数 κ 是径向基函数，那么功率谱密度为 $p(\boldsymbol{\omega}) = \mathcal{N}(\mathbf{0}, \Lambda^{-1})$，其中 $\Lambda = \text{diag}\{\lambda_1^2, ..., \lambda_{D_x}^2\}$。

由于协方差函数和非负测度均为实数，我们舍弃期望自变量中的复数部分，并保留 $\cos\left(\boldsymbol{\Delta}^\top \boldsymbol{\omega}\right) = \cos\left((\mathbf{x} - \mathbf{x}')^\top \boldsymbol{\omega}\right)$，它也可以表示为 $\cos(\mathbf{x}^\top \boldsymbol{\omega}) \cos(\mathbf{x}'^\top \boldsymbol{\omega}) + \sin(\mathbf{x}^\top \boldsymbol{\omega}) \sin(\mathbf{x}'^\top \boldsymbol{\omega})$。接下来，我们注意到，利用上述展开，可以通过蒙特卡洛采样估计 $[\cos(\mathbf{x}^\top \boldsymbol{\omega}) \; \cos(\mathbf{x}'^\top \boldsymbol{\omega})]$ 来近似 $\kappa(\mathbf{x}, \mathbf{x}')$。若 $\mathbf{z}(\mathbf{x}) = [\cos \mathbf{x}, \sin \mathbf{x}]$，则可以得到 $(\mathbf{x}, \mathbf{x}')$ 的无偏估计：

$$\overline{\kappa}(\mathbf{x}, \mathbf{x}') \approx \frac{1}{J} \sum_{j=1}^{J} \mathbf{z}(\mathbf{x}, \boldsymbol{\omega}_j) \mathbf{z}(\mathbf{x}', \boldsymbol{\omega}_j)^\top, \tag{7}$$

其中，$\boldsymbol{\omega}_j$s 是从 $p(\boldsymbol{\omega})$中抽取的样本，J 是谱频率的随机样本数。利用$\mathbf{x}$ 和随机样本 $\boldsymbol{\omega}_j$（它们是$\boldsymbol{\Omega}$的列），并且$\boldsymbol{\Omega} \in \mathbb{R}^{D_x \times J}$（或者，$\boldsymbol{\Omega}_{\cdot j} = \boldsymbol{\omega}_j$），我们将 RBF 核函数的随机特征定义为

$$\boldsymbol{\phi}_{\text{rbf}}(\mathbf{x}) \triangleq \frac{\sigma_\lambda}{\sqrt{J}} [\sin(\mathbf{x}^\top \boldsymbol{\Omega}), \cos(\mathbf{x}^\top \boldsymbol{\Omega})]^\top, \tag{8}$$

其中

$$\sin(\mathbf{x}^\top \boldsymbol{\Omega}) = \left[\sin\left(\mathbf{x}^\top \boldsymbol{\omega}_1\right) \; \sin\left(\mathbf{x}^\top \boldsymbol{\omega}_2\right) \; \ldots \sin\left(\mathbf{x}^\top \boldsymbol{\omega}_J\right)\right] \tag{9}$$

$\cos(\mathbf{x}^\top \boldsymbol{\Omega})$ 的定义与此类似。

对于 ARC-COSINE 核，我们可以写作 [24]

$$\begin{aligned}
&\widetilde{\kappa}(\mathbf{x}, \mathbf{x}') \\
&= 2 \int \mathbf{x}^\top \boldsymbol{\omega} \; \boldsymbol{\omega}^\top \mathbf{x}' \, H(\boldsymbol{\omega}^\top \mathbf{x}) H(\boldsymbol{\omega}^\top \mathbf{x}') \mathcal{N}(\boldsymbol{\omega} | \mathbf{0}, \mathbf{I}) \mathrm{d}\boldsymbol{\omega} \\
&\approx \frac{2}{J} \sum_{j=1}^{J} \mathbf{x}^\top \boldsymbol{\omega}_j \; \boldsymbol{\omega}_j^\top \mathbf{x}' \, H(\boldsymbol{\omega}_j^\top \mathbf{x}) H(\boldsymbol{\omega}_j^\top \mathbf{x}').
\end{aligned} \tag{10}$$

其中，$H(\cdot)$ 是 Heaviside 阶跃函数。与 RBF 核类似，基于此协方差的随机特征定义为 [29]

$$\boldsymbol{\phi}_{\text{arc}}(\mathbf{x}) \triangleq \frac{\sqrt{2}\sigma_\lambda}{\sqrt{J}} \max\left(\mathbf{0}, \boldsymbol{\Omega}^\top \mathbf{x}\right). \tag{11}$$

我们注意到，恒等特征为

$$\boldsymbol{\phi}_{\text{I}}(\mathbf{x}) \triangleq [1 \; \mathbf{x}^\top]^\top. \tag{12}$$

当我们采用式（12）中的特征空间时，我们的模型就简化为普通的贝叶斯层次混合专家模型。因此，我们将核函数的不同近似表示为 $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$，并将 f 的高斯过程近似定义为

$$\widehat{f}(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x})^\top \mathbf{w}, \tag{13}$$

其中，权重$\mathbf{w}$ 是一个具有高斯先验的参数向量，$\mathcal{N}(\mathbf{0}, \mathbf{I})$。

3 高斯过程门控分层专家混合模型

我们提出了使用小批量梯度下降训练的高斯过程门控分层专家混合模型（GPHME），以构建专家混合模型。图 1 展示了一个 GPHME 示例。树中的每个内部节点ν 都有一组学习得到的系数$\mathbf{w}_\nu$ 和一个随机向量$\boldsymbol{\Omega}_\nu$，而每个叶节点 l 都有一个学习得到的分布 Q_l。在每个内部节点，选择最左侧的分支为 $\sigma(z_\nu)$，其中 $\sigma(\cdot)$ 是 Sigmoid 函数。变量 z_ν 定义为

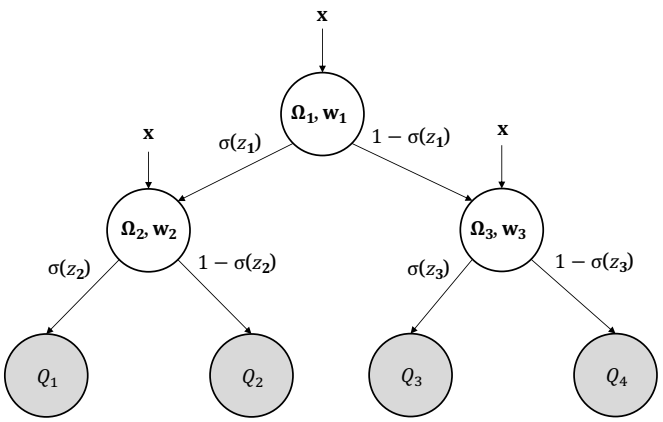

图 1. 具有固定树结构的 GPHME，由专家叶节点（以阴影圆圈表示）和内部节点（以圆圈表示）组成。边表示与内部节点相关联的基于随机森林的决策规则，而 Qs 表示关于目标变量 y的条件分布。

$$z_\nu(\mathbf{x}) = \boldsymbol{\phi}(\mathbf{x}^\top \boldsymbol{\Omega}_\nu) \mathbf{w}_\nu, \tag{14}$$

其中 $\mathbf{x}$ 是模型的输入，$\boldsymbol{\phi}$ 是随机特征。因此，节点 ν 处的 GP 门控模型取决于 z_ν，后者是随机特征 $\boldsymbol{\phi}$ 的函数。显然，门控概率 $\sigma(z_\nu)$ 是先计算 z_ν，再通过 $\sigma(\cdot)$ 将其映射到 $[0, 1]$ 得到的。公式（14）给出了输入空间的非线性划分。

图 1 所示的模型是一种混合专家模型，它学习系数层次结构，用于根据各自的路径概率将每个样本分配给 GP专家。与 [12], 不同，在 [12], 中，每个专家实际上都是一个“偏执者”，因为它不会使用测试数据来计算专家的概率分布；而我们模型中的专家则根据 softmax函数，学习给定测试输入$\mathbf{x}$ 时可能输出类别上的分布

$$Q_l^k = \frac{\exp(z_l^k)}{\sum_{k'} \exp(z_l^{k'})}, \tag{15}$$

其中每个$(\mathbf{x}^\top \boldsymbol{\Omega}_l) \mathbf{w}_l^k$ 是一个已学习的概率，表示第 l个叶节点上的类别 k，但不包括归一化常数；并且

Q_l^k 表示第 l个叶节点分配给类别 k 的概率。我们将隐变量统称为$\mathbf{\Theta} = \{\mathbf{w}_\nu、\ \mathbf{w}_l、\ \mathbf{\Omega}_\nu、\ \mathbf{\Omega}_l\}$，其中索引 ν指内部节点，l 指叶节点。目标函数，即输入-输出对($\mathbf{x}$ y)的 $\mathbf{\Theta}$ 混合对数似然，定义为

$$\begin{aligned}\log p(y|\mathbf{x},\mathbf{\Theta}) &= \log \sum_l P(l|\mathbf{x},\mathbf{\Theta}) p(y|\mathbf{x},l,\mathbf{\Theta}) \\ &\geq \sum_l P(l|\mathbf{x},\mathbf{\Theta}) \log p(y|\mathbf{x},l,\mathbf{\Theta}) \\ &=: \log \widetilde{p}(y|\mathbf{x},\mathbf{\Theta}),\end{aligned} \tag{16}$$

其中，$P(l|\mathbf{x}\ \mathbf{\Theta})$ 表示给定输入 $\mathbf{x}$ 和 $\mathbf{\Theta}$ 时到达叶节点l 的概率，而 $p(y|\mathbf{x}\ l,\mathbf{\Theta}) = Q_l^y$ 表示第 l 个专家的 $\mathbf{\Theta}$ 似然。为缓解数值计算方面的挑战，我们选择在 ELBO 优化中优化 $\log \widetilde{p}(y|\mathbf{x}\ \mathbf{\Theta})$，而非 $\log\ p(y|\mathbf{x}\ \mathbf{\Theta})$。具体而言，当每个叶节点 l 对应的 $P(l|\mathbf{x}\ \mathbf{\Theta})$ 和 $p(y|\mathbf{x}\ l,\mathbf{\Theta})$ 都趋近于零时，$\log\ p(y|\mathbf{x}\ \mathbf{\Theta})$ 将趋向负无穷。对于多元回归任务，叶节点只需通过 $Q_l^k = z_l^k$ 即可提供回归器，其中 k 指多元 $\mathbf{y}$ 的维度。我们再次强调，该模型本质上是 GP 专家的层次混合：我们将输入 $\mathbf{x}$ 投影到不同的特征空间，以确定从根节点到叶节点的路径，并在叶节点预测目标分布。

然而，式（16）中的目标函数会促使每个叶节点最小化其对数似然，进而导致每个叶节点都成为擅长所有类别的专家，因此不会偏向任何类别。换言之，这些专家会变得样样通、样样松。尽管该目标函数有助于获得更准确的结果，但为了让结果更具可解释性，我们或许希望每个叶节点都偏好某个特定类别。因此，目标函数的另一种选择是归一化似然。令

$$\begin{aligned}\log Q^k &= \sum_l P(l|\mathbf{x},\mathbf{\Theta}) \log p(k|\mathbf{x},l,\mathbf{\Theta}) \\ &\geq \sum_l P(l|\mathbf{x},\mathbf{\Theta}) \log p(k|\mathbf{x},l,\mathbf{\Theta}) \\ &=: \log \widetilde{Q^k},\end{aligned} \tag{17}$$

与给定$\mathbf{x}$时类别 k的对数概率分布成正比。具体而言，(16)等价于$\log Q^y$。与(16)之后的讨论一致，我们使用$\log\ Q^k$而非$\log Q^k$，归一化似然定义为

$$\text{Normalized Likelihood} = \frac{\widetilde{Q^y}}{\sum_k \widetilde{Q^k}}. \tag{18}$$

换句话说，我们不仅最大化类别 y 的似然，还会考虑 y 相对于其他类别的似然。因此，训练后，每个专家都会偏好一个类别，也就是说，相对于其他类别，它平均而言会以更高的概率识别出某个特定类别。我们将式（16）中的目标函数称为目标函数一（OF1），将式（18）中的目标函数称为目标函数二（OF2）。我们在这两个函数中分别看到了线性汇聚函数和对数线性汇聚函数。需要指出的是，多个专家可能都对应同一个类别，并且它们是自动训练的。

4 变分推断

我们的目标是找到一个变分分布 q（$\mathbf{\Theta}$），使其近似真实后验分布 p（$\mathbf{\Theta}$|$\mathbf{X}$，$\mathbf{Y}$）。定义边缘化对数似然 $L = \sum_{n=1}^N\ \ p\ y|\ \ \ \ L' = \sum_{n=1}^N \mathbb{E}_{q(\mathbf{\Theta})}\ \ \ p\ y|\ \ \mathbf{\Theta}\ \geq \log \sum_{n=1}^N \mathbb{E}_{q(\mathbf{\Theta})}\ \ \widetilde{p}\ y|\ \ \mathbf{\Theta}\ =\ L''$ （$\mathbf{x}$） 和 log（$\mathbf{x}$） log（$\mathbf{x}$）：，我们有

$$L \geq L' - \mathrm{KL}\left[q(\mathbf{\Theta})||p(\mathbf{\Theta})\right] \geq L'' - \mathrm{KL}_{\boldsymbol{\lambda}}\left[q(\mathbf{\Theta})||p(\mathbf{\Theta})\right], \tag{19}$$

其中 KL 表示 Kullback-Leibler散度，$p(\mathbf{\Theta})$ 是隐变量 $\mathbf{\Theta}$的先验分布，其中

$$p(\mathbf{\Theta}) = \prod_\nu \left[p(\mathbf{w}_\nu) \prod_j p(\boldsymbol{\omega}_{\nu j}) \right]. \tag{20}$$

通过优化证据下界，先验 $p(\boldsymbol{\omega})$ 中的超参数也会得到优化。注意，KL散度会自动对 $\mathbf{\Theta}$ 进行正则化，从而在$||\mathbf{w}_\nu||$过大时避免过拟合。

我们假设一个在各节点间分解的高斯近似分布。那么我们有

$$q(w_{\nu j}) \sim \mathcal{N}(m_{\nu j}, (s_{\nu j})^2), \tag{21}$$

$$q(\Omega_{\nu ij}) \sim \mathcal{N}(\mu_{\nu ij}, (\sigma_{\nu ij})^2), \tag{22}$$

其中，$w_{\nu j}$ 表示$\mathbf{w}_\nu$的第 j个元素，而$\mathbf{\Omega}_{\nu ij}$ 是$\mathbf{\Omega}_\nu$ 中第 i行、第 j列的元素。变分参数是每个近似因子 $m_{.j}$、$s_{.j}$、$\mu_{.ij}$和 $\sigma_{.ij}$的均值与方差，我们旨在针对这些参数优化下界。内部节点$\mathbf{w}_\nu$ 和$\mathbf{\Omega}_\nu$ 的设置与叶节点$\mathbf{w}_l^k$和$\mathbf{\Omega}_l$类似。由于观测相互独立，我们对 L''采用双随机近似。如果随机选取由 $\mathcal{I}_M$索引的 M 个点，则可使用小批量以无偏方式估计 L''：

$$L'' = \frac{N}{M} \sum_{m \in \mathcal{I}_M} \mathbb{E}_{q(\mathbf{\Theta})} L(y_m|\mathbf{x}_m,\mathbf{\Theta}). \tag{23}$$

为处理期望项，我们采用蒙特卡罗采样，得到

$$L'' = \frac{N}{M} \sum_{m \in \mathcal{I}_M} \frac{1}{N_{MC}} \sum_{r=1}^{N_{MC}} L(y_m|\mathbf{x}_m,\mathbf{\Theta}_r), \tag{24}$$

其中，$\mathbf{\Theta}_r$ 从 $q(\mathbf{\Theta})$中采样，N_{MC} 是抽取的样本数。为了利用反向传播，我们应用重参数化技巧，将权重重新参数化如下：

$$w_{\nu jr} = m_{\nu jr} + s_{\nu jr}\epsilon_{\nu jr}, \tag{25}$$

$$\Omega_{\nu ijr} = \mu_{\nu ijr} + \sigma_{\nu ijr}\epsilon_{\nu ijr}, \tag{26}$$

其中，$\epsilon_{\nu jr}$和$\epsilon_{\nu ijr}$是来自标准正态分布的独立样本。也可以为 σ_λ 和 λ_d 指定先验分布，然后使用变分推断。

本质上，带有高斯过程学习器的叶节点彼此竞争。输入 $\mathbf{x}$ 往往会沿着高斯过程学习器更有效的叶节点所在路径前进。梯度下降方法会加剧这种倾向，使学习

过程进入这样一种状态：只有一个叶节点显著优于其他节点，输入也倾向于以接近 1 的概率落入该叶节点——这与“赢家通吃”情形类似。为解决这一问题，我们采用了 [12], 提出的方案，通过惩罚路径，鼓励每个内部节点均衡地使用左右子树。我们将$\mathbf{x}$ 在节点 ν处转向左子节点的平均概率定义为

$$\alpha_\nu = \frac{\sum_{\mathbf{x}} P_\nu(\mathbf{x}) p_\nu(\mathbf{x})}{\sum_{\mathbf{x}} P_\nu(\mathbf{x})}, \tag{27}$$

其中， $P_\nu(\mathbf{x}) = P(\nu|\mathbf{x}\ \boldsymbol{\Theta})$ 表示$\mathbf{x}$ 到达节点 ν 的概率，而 $p_\nu(\mathbf{x})$ 表示$\mathbf{x}$ 到达节点 ν的左子节点的概率。因此，惩罚项变为

$$C = \lambda N \sum_\nu \left[0.5 \log(\alpha_\nu) + 0.5 \log(1-\alpha_\nu)\right], \tag{28}$$

其中， λ 是一个超参数，在我们的实验中设为 2^{-d} 。惩罚证据下界（PELBO）定义为

$$PELBO = L'' + C - \mathrm{KL}\left[q(\boldsymbol{\Theta})||p(\boldsymbol{\Theta})\right]. \tag{29}$$

到目前为止，我们将不同节点上的$\boldsymbol{\Omega}_\nu$ 设为不同值，即每个节点都有唯一的随机特征空间（NIS-N），因此这些空间是非各向同性的。不过，这可能并非必要，而且会造成巨大的空间复杂度开销。我们可以设想存在一个特定的随机特征空间，能够将数据线性分离，这意味着所有节点共享$\boldsymbol{\Omega}$的同一分布。我们将此选项称为所有节点的各向同性空间（ISO-N）。另一种做法是限制每一层的节点，使其共享$\boldsymbol{\Omega}$ 的同一分布，从而减轻计算负担。我们将此称为同层节点空间的各向同性（ISO-L）。然而，这种安排似乎并不合理，因为很难证明将同一层中彼此相距甚远的节点合并是合理的。在数值实验一节中，结果确实表明，这一选项在大多数情况下都不如其他选项。实践中，ISO-N 选项略逊于 NIS-N。不过，尽管性能稍差，ISO-N 选项由于采用单一的投影特征空间，因而比 NIS-N 选项具有更好的可解释性。

推断的两个主要操作来自公式 (14) 和 (15)。对于大小为 M 的一个小批量，公式 (14) 的计算复杂度为 $\mathcal{O}(MJD_xN_{MC}) + \mathcal{O}(MJN_{MC})$，而公式 (14) 的计算复杂度为 $\mathcal{O}(MJD_xN_{MC}) + \mathcal{O}(MJD_yN_{MC})$。结合树的高度 h，最终的训练复杂度为 $\mathcal{O}(2^hMJN_{MC}(D_x + D_y))$。测试复杂度为 $\mathcal{O}(2^hN_{test}JN_{MC}(D_x + D_y))$，其中 N_{test} 是测试集的大小。

本文所用结果的代码可在 GitHub 上获取。[1]

5 数值实验

本节首先展示我们的 GPHME 如何通过模仿 DGP 的行为来解释它们。随后，我们讨论$\boldsymbol{\Omega}$ 的不同设置以及树的高度选择。接下来，我们将我们的方法与贝叶斯混合专家模型（BHMEs） [2] 以及 Soft 树和硬树进行比较。此外，我们还将模型与基于树的高斯过程的最先进水平进行比较。所有结果都表明，有必要在特征空间中进行投影。最后，我们展示了将我们的方法应用于大规模数据的实现结果。

所有实验均在一台配备 24GB 随机存取存储器的 NVIDIA TITANRTX GPU 单机上运行，但也可以直接在中央处理器上运行。

5.1 GPHME 的可解释性与可解释说明

我们将可解释性理解为借助树状结构来理解更复杂的模型，例如深度高斯过程和深度神经网络。在我们的方法中，节点所做的每项决策——选择遍历左子树还是右子树——都对应一个高斯过程。因此，从根节点到叶节点的路径由一系列决策构成，这些决策共同形成一个深度高斯过程或深度神经网络。在我们的框架中，我们将可解释说明定义为解释分类结果的能力。

图 2 展示了我们的方法如何对 MNIST 数据作出决策。以最左侧的四个叶节点为例，可以看到，最可能的分类结果是 4 和 9，因此它们的父节点只需学习区分这两个数字即可。对于 4 和 9 这两组而言，这样的划分是合理的，因为这两个数字都包含封闭区域。此外，2 和 5 是最难分类的数字，因为似乎没有任何专家能给出足够高的概率，以便作出有把握的判断。内部节点的决策方式与此类似，也很容易理解。总之，我们模型的可解释说明包含两个关键方面：

- 我们设计的归一化似然会引导每个叶节点尽可能偏向某一个类别，从而解释相应叶节点高斯过程的功能。
- 由于叶节点高斯过程主要偏向某一个类别，树结构便解释了哪些类别彼此更相似。

在具备良好可解释性的同时，训练期间，我们的模型在树高为四且采用 ISO-N 设置时达到最高准确率 97.79%。若使用式 (16) 中的损失函数，则准确率可达 98.49%，仍是在 ISO-N 设置下。尽管参数更多，深度高斯过程模型仅在隐藏层数为一时达到峰值 98.04%，且随着层数增加，准确率会下降。我们的 GPHME 模型也可与单隐藏层深度神经网络媲美，后者的准确率为 98.4% [30]。GPHME 达到这一准确率时，参数数量仅为深度神经网络的三分之二。此外，在 NIS-N 设置下，GPHME 在树高仅为二时，准确率最高可达 98.67%，与支持向量机报告的 98.6% 相当 [31]。它也优于其他基于核的方法，包括高斯过程及其变体 [32], [33]。

5.2 超参数讨论

由于特征空间中嵌入了信息，我们的基于树的模型无需将树构建得太深。因此，本节将树的高度设为不超过两层。我们选取了一个回归数据集 $Protein$ 和一个

1. https://github.com/yuhaoliu94/GP-HME.

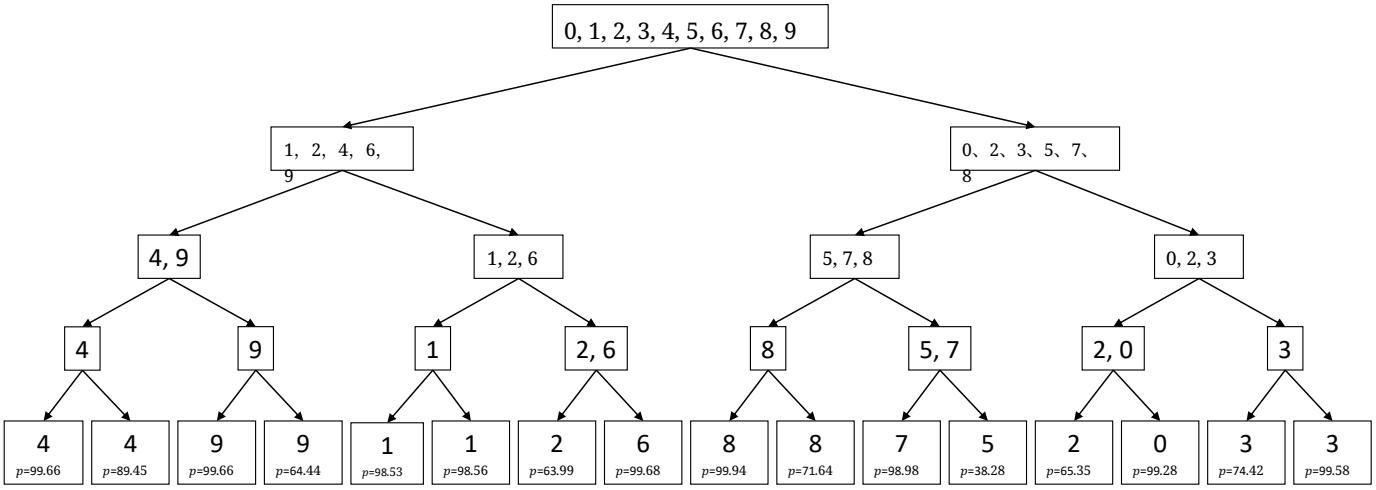

图2. 在 MNIST 数据集上训练的深度为四的 GPHME 可视化图。每个叶节点处显示最终最可能的分类结果及其在样本上的平均概率。每个内部节点标注的类别都是从叶节点向根节点反向追溯得到的。叶节点并非只预测一个类别，而是预测所有类别。例如，如果有 100 个样本，某个叶节点将其中 80 个预测为数字 0，将另外 20 个预测为数字 1–9 中的任意数字，那么图中的该叶节点就标注为数字 0，概率为 =0.8。内部节点中写出的类别是从叶节点开始，逐层向上推导得出的，因为预测发生在叶节点处。我们的模型是一种“软”决策树，路径具有概率，因此一个数字可能同时“走向”左侧和右侧。

多输出分类数据集 $Optical\ \ Digits$ (OPT)，来自 UCI 仓库。[2] 径向基函数核和 ARC-COSINE 核的随机特征维度固定为 100。图4展示了不同设置下的性能，包括不同的树高、**Ω**结构以及核类型。为简洁起见并使图表易于阅读，我们仅展示树高为两层时不同**Ω** 结构的结果。

图3展示了模型在不同树高和不同**Ω** 选项下的工作情况。该图还展示了树高为一时，不同**Ω** 选项下的结果。

正如我们预期的那样，随着树的高度增加，性能并不总是有所提升，因为这会导致过拟合。对于 **Ω** 的设置，NIS-N 选项的参数数量最多，但与 ISO-L 和 ISO-N 选项相比，结果并未显著改善。在某些情况下，ISO-N 选项优于 NIS-N 选项。这也是我们在实践中更倾向于使用 ISO-N 选项的原因之一。ISO-L 选项不够稳定，在所有 **Ω** 设置中可能得到最差的结果。这是合理的，因为该选项会强制同一层级的节点共享相同的特征空间。至于核函数的选择，RBF 核通常表现更好，而 ARC-COSINE 核在二分类任务中可能效果更佳。下一节将讨论恒等特征，并将其作为一种基准模型。

5.3 UCI 数据集

为评估各种基于树的模型的泛化误差和模型复杂度，包括普通决策树（线性判别树 - LDT、硬树 - C4.5 和软树），以及基于树的高斯过程（贝叶斯加性回归树 - GP-BART、贝叶斯树状高斯过程 - TGP 和 BHMEs），我们采用了与以下研究相同的数据集

2. https://archive.ics.uci.edu

该研究使用了这些数据集，并报告了 LDT、C4.5 和软决策树的结果。不过，目前有些数据集已无法获取。因此，我们只选取了仍然可用的数据集，包括四个回归数据集（ABAlone、ADD10、BOSton、CONcrete）、八个二分类数据集（BREast、GERman、MAGic、MUSk2、PIMa、RINgnorm、SPAmbase、TWOnorm）和十个多分类数据集（BALance、CMC、DERmatology、ECOli、GLAss、OPTdigits、PAGeblock、PENdigits、SEGment、YEAst），这些数据集均来自 UCI 仓库。基准方法采用五折数据划分，并对结果取平均；其中三分之一的数据作为测试数据，其余三分之二作为训练数据。我们在此采用相同的设置，并将模型的树高限制为最多两层。由于 GP-BART [27]和 TGP [26]专为回归任务设计，我们使用回归数据集评估它们的性能。我们确保这两个模型的预热期和 MCMC 迭代总数保持一致，并遵循 GP-BART 的默认设置。为公平地与 BHMEs 比较，我们将 BHMEs 的树高设得足够大，使其参数数量略多于我们的模型。在这种情况下，BHMEs 的树高通常平均达到 7 或 8 层，尽管其在树高为两层或三层时几乎总能取得最佳结果。此外，我们报告了五个随机打乱的折之间的标准差，而决策树基准方法的原始文献未报告这一数值。结果取自不同树高和核类型中表现最佳的模型。

表 1、2、3 展示了回归任务的均方误差、训练时间和树大小。表 4 和 5 展示了二分类和多分类的准确率。这些表还列出了样本数 N、输入维度 D_x 和输出维度 D_y。表 6 和 7 还列出了分类任务在五个不同折上的平均训练时间。训练过程的时长设定为不超过 60 分钟。

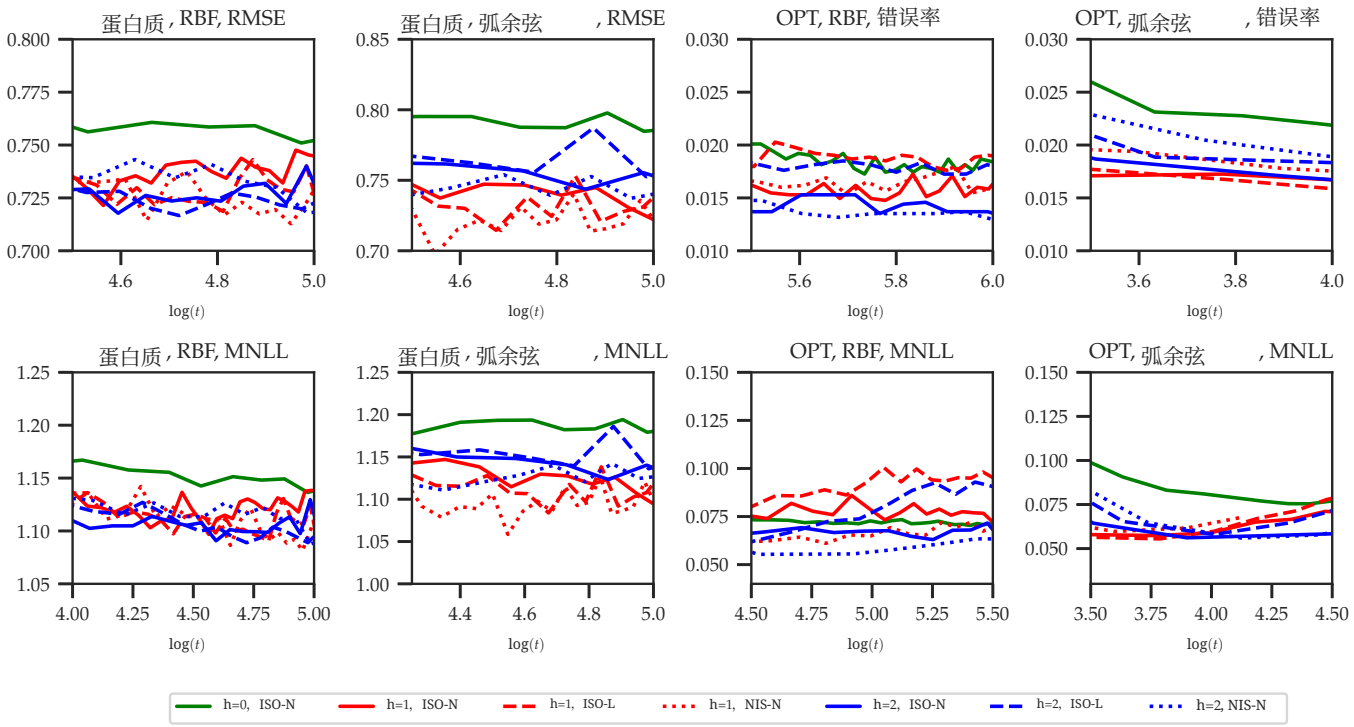

图3. 回归情况下均方根误差随时间的变化 回归情况下的错误率，分类情况下的错误率，以及平均负对数似然（MNLLs）ove r 时间变化。

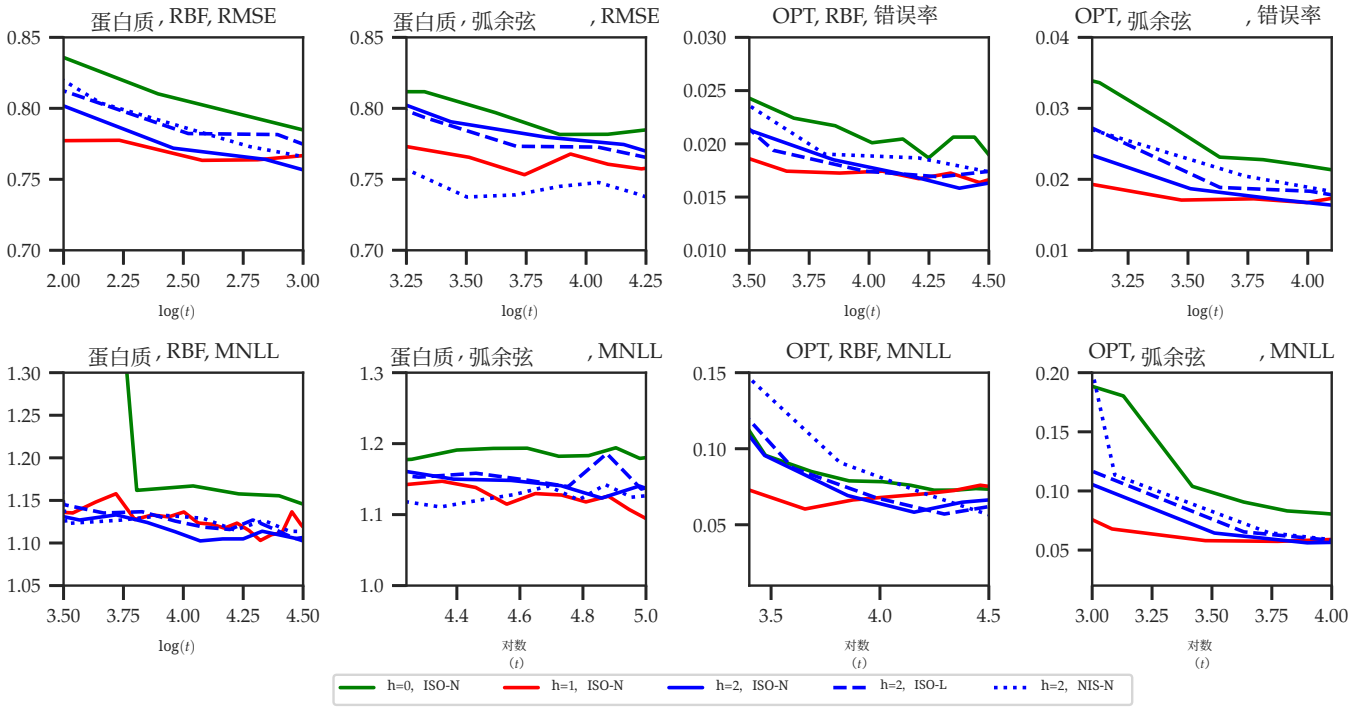

图4. 回归情况下均方根误差、分类情况下的错误率，以及平均负对数似然（MNLL）随时间的变化。平均负对数似然图的 x 轴与均方根误差图和错误率图的 x 轴不同，因为平均负对数似然比均方根误差和错误率更晚收敛。

所用树的大小由所有内部节点和叶节点的数量指定。其他常规决策树的结果包括 C4.5（对应单变量硬树）以及 LDT，它是一种斜向多变量硬树。结果还提供了基于树的高斯过程的其他基准方法，包括 GP-BART 和 TGP，两者均采用确定性线性门控规则。由此产生的确定性门控决策导致

叶节点将输入空间划分为多个区域，从而促使每个叶节点在其对应的输入区域内训练一个高斯过程。

首先，我们将本方法的结果与用于回归的基于树的高斯过程进行比较，即 GP-BART 和 TGP。我们注意到，在模型结构和推断方面，我们的模型优于 GP-BART 和 TGP。这些基于树的高斯过程基准方法使用精确高斯过程，需要计算高维矩阵的逆。当叶节点包含大量样本时，这种方法就不切实际，训练时间也会变得难以接受。因此，这些方法最适用于数据集较少的情况。此外，它们采用纯马尔可夫链蒙特卡洛方法进行推断，虽然精度较高，但执行速度较慢。再者，尽管这些基于树的高斯过程基准方法主要侧重于解释路径，我们的模型更进一步，还能解释叶节点上的模型。

我们对所有模型进行了不等方差（Welch）t 检验。加粗的结果表示，本方法显著优于其他模型中的一半。从表1可以看出，在 CON 数据上，无论是均方误差还是训练时间，我们的模型都优于基准方法。考虑到基准方法在理论上因采用马尔可夫链蒙特卡洛方法而预期会更加精确，这一点尤为值得注意。参见表2，基准方法难以高效地训练大型数据集。对于 ABA 和 ADD，基准方法运行一次迭代约需一分钟。默认设置要求预热期之前进行 1,500 次迭代，并在完成前再进行 3,500 次迭代，总耗时超过两天。请注意，表3中的树大小无法从 GP-BART 模型的输出中获得。

从表 4 和表 5 的结果可以看出，在损失方面，我们的模型优于所有其他基于树的模型。例如，对于 RIN 数据集，BHME 候选模型的准确率仅约为 77%，而对于相同规模的树，我们的模型将准确率提高到了98% 以上。此外，在约 80% 的情况下，我们的树比基准树更小。由于普通决策树模型 [34] 的估计方差未提供，因此我们未将其纳入 Welch 检验。基准结果来自 [10], [34] 中报告的树。不过，这些论文的作者并未提供估计方差。

5.4 大规模数据集

GPHME 可以视为弱学习器，并可为 DGP 或更广泛而言的贝叶斯深度神经网络提供可解释性。我们原本并不指望自己的模型优于 DGP，而只是想用它来解释 DGP。不过，令我们惊讶的是，在大多数时候，我们的模型表现优于（至少略微优于）DGP。此外，我们模型的一个显著特点是能够扩展至大型数据集。

我们在两个大规模问题上评估了模型，其规模超出了高斯过程，尤其是深度高斯过程通常应用的数据集规模。首先，我们考察了 MNIST8M，它将原始 MNIST 数据集人为扩展至 8+百万个观测值。我们采用标准 MNIST 实验所述的相同配置训练该模型，树的高度设为二，并使用 NIS-N 设置。我们在测试集上取得了 99.30% 准确率，MNLL 为 0.0372；这些结果优于 [29] 给出的 DGP 对照结果。值得指出的是，我们模型的参数数量少于深度高斯过程。请注意，[33] 最近采用 AutoGP 框架取得了 99.11% 的准确率，MNLL 为 0.033；而我们的模型取得了最低的 MNLL（0.0328），准确率为 99.24%。

GP 领域常见的大规模数据集之一是 AIR- LINE 数据集，其中包含 2008年 5+百万次美国航班的航班信息。尽管该数据集并未公开，我们找到了一份替代性的公开数据集，其中包含超过 6+ 百万条航班信息记录。[3] 我们使用这个 8 维数据集进行分类，以确定航班是否延误。我们使用 [35]，中提供的脚本构建测试集，并留出 100,000 个数据点用于测试。我们使用 100 个基于径向基函数的随机特征构建模型，并将树的高度设为 1，使其参数数量与单隐藏层深度高斯过程 [29] 相匹配。如表 8 所示，我们的模型取得的结果与深度高斯过程的结果直接相当，这意味着决策树几乎可以达到深度高斯过程的性能。此外，当我们将树的高度增加到 4 时，模型的准确率达到 72%，而具有 10 个隐藏层的对应深度高斯过程却无法收敛。

MNIST 数据的训练时间为 40 分钟，航空公司数据的训练时间为 19 分钟。

另一种适用于大规模数据集的基准方法是 GP-tree 模型 [28]。该模型采用基于诱导点的高斯过程作为门控模型。遗憾的是，目前尚无法成功实现这一基准方法的代码。虽然目前无法实现 GP-tree 模型的代码，但值得注意的是，我们的模型相较于这一基准 GP-tree 模型仍然具有优势。

我们再补充两点说明。第一，GP-tree 模型仅专注于图像分类，并依赖卷积神经网络进行特征提取。相比之下，我们的模型用途更广，既能处理回归任务，也能处理一般的分类任务。第二，GP-tree 模型会预先指定叶节点的类别标签，以确保可解释性，因此需要先验信息来将类别划分为不同组别。相比之下，我们的模型会自动引导每个叶节点倾向于单一类别。

6 总结

本文提出了一种新颖的分层专家混合模型，其内部节点和专家均由高斯过程表示。我们选择采用随机特征来扩展高斯过程。我们的 GPHME 优于所有基准方法，包括树结构分层专家混合和决策树。与其他基于高斯过程的分层专家混合相比，GPHME 的复杂度更低，并且能够为深度高斯过程和深度贝叶斯神经网络提供可解释性。分层专家混合模型的一个局限是需要预先选择树的规模，而且其参数数量会呈指数级增长。然而，实践表明，我们并不需要高度很大的树。我们在各种数据集上的结果清楚地表明，即使使用较小的树，在大规模数据集上也能取得出色的性能。

未来的工作包括以下内容：

3. ”https://www.kaggle.com/vikalpdongre/us-flights-data-2008”

1）剪枝。训练完成后，可以对树进行剪枝。例如，图 2 显示，我们可以合并预测相同类别的子树。研究有原则的剪枝方法十分重要。2）扩展到 GPHME 集成。假设我们有多个 GPHME，每个都由各自的一组特征空间定义。此外，假设这些 GPHME 的先验信息表明，其树的结构属于 ISO-N，即 $\mathbf{\Omega}$ 在所有节点上相同，且它们具有已知的特征空间$\mathbf{\Phi}$。在这种情况下，输入空间被投影到不同的固定特征空间，而不是随机特征空间。与随机特征空间相比，固定特征空间更容易解释。因此，决策树集成自然就是一种利用已知固定特征空间的集成。重要的是探索识别优质集成的方法，例如允许树随时间变化，或利用比各自先验更具信息量的谱频率分布函数，从性能良好的 GPHME 中生成新树。3）扩展到提升。这项任务相当直接，因为我们只需在构建一系列树时复现标准的提升流程。4）用深度高斯过程对专家进行建模。将树的专家（叶节点）从目前使用的高斯过程替换为深度高斯过程，或许可以提升 GPHME 的性能。这样做既不会失去 GPHME 良好的可解释性，又可能通过使用深度高斯过程作为专家提升性能。5）特征选择。传统树可以提供关于$\mathbf{X}$最重要特征的信息。我们的树是否也能提供重要特征的信息？我们计划研究能否从$\mathbf{\Omega}$和$\mathbf{w}$中提取特征重要性信息。6）扩展到诱导点近似。利用诱导点，我们有另一种实现可扩展高斯过程的方法。研究基于此类高斯过程的树与本文所述树的比较结果将很有意义。7）前馈选项。本文中，我们仅使用特征空间 $\boldsymbol{\phi}(\mathbf{x})$ 进行决策和预测。然而，一个有趣的方向是探索使用前馈选项来构建树，即同时使用 $\boldsymbol{\phi}(\mathbf{x})$ 和 $\mathbf{x}$。目标不仅是提高树执行任务时的准确率，还要了解加入前馈选项如何影响树的结构及其参数。

回归任务的均方误差

	N	D_x	GPHME	BHME	GP-BART	TGP	Soft	Hard
ABA	4177	8	0.405 ± 0.016	0.414 ± 0.014	-	-	0.439	0.557
ADD	9792	10	**0.043 ± 0.001**	0.052 ± 0.003	-	-	0.094	0.267
BOS	506	13	0.112 ± 0.023	0.112 ± 0.020	0.114 ± 0.043	0.101 ± 0.014	0.094	0.267
CON	1030	8	**0.098 ± 0.003**	0.147 ± 0.012	0.118 ± 0.019	0.101 ± 0.010	0.264	0.286

表2 回归任务的训练时间（分钟）

	N	D_x	GPHME	BHME	GP-BART	TGP
ABA	4177	8	10.7126	42.9456	>1天	>1 天
ADD	9792	10	12.1984	59.9985	>1天	>1 天
BOS	506	13	14.8210	45.3653	2.5596	2.0336
CON	1030	8	12.0774	59.8728	22.2742	15.0907

表3 回归任务的树大小

	N	D_x	GPHME	BHME	GP-BART	TGP（5折中的最大值）Soft		Hard
ABA	4177	8	3	15	-	-	7	32
ADD	9792	10	3	127	-	-	15	202
BOS	506	13	3	15	-	3	11	18
CON	1030	8	3	31	-	3	13	69

表4 二分类任务的准确率

			准确率（%）					树大小				
	N	D_x	GPHME	BHME	Soft	C4.5	LDT	GP.	BH.	Soft	C4.5	LDT
BRE	569	30	98.42 ± 00.88	97.47 ± 00.91	95.34	93.80	95.09	3	15	17	47	4
GER	1000	24	76.77 ± 01.42	76.71 ± 01.19	75.74	69.07	74.16	7	7	16	142	7
MAG	19020	10	**87.83 ± 00.22**	79.25 ± 00.29	81.27	84.09	83.07	7	63	17	1072	40
MUS	6598	167	100.00 ± 00.00	100.00 ± 00.00	92.25	94.62	93.59	3	3	22	202	15
PIM	768	8	78.52 ± 00.96	78.13 ± 01.44	70.85	69.41	76.89	3	15	26	111	5
RIN	7400	20	**98.05 ± 00.16**	76.71 ± 00.62	88.94	87.54	77.25	7	7	368	354	4
SPA	4601	57	**94.85 ± 00.41**	92.95 ± 00.48	78.38	90.14	89.86	3	3	22	155	13
TWO	7400	20	98.01 ± 00.24	98.06 ± 00.21	97.92	87.59	98.00	7	3	41	429	3

表5 多个分类任务的准确率

				准确率(%)					树大小				
	N	D_x	D_y	GPHME	BHME	Soft	C4.5	LDT	GP.	BH.	软 C4.5		LDT
BAL	625	4	3	**97.80 ± 00.72**	90.43 ± 02.40	89.85	61.91	88.46	3	7	10	5	3
CMC	1473	9	3	100.00 ± 00.00	100.00 ± 00.00	52.03	50.00	46.64	3	3	21	24	3
DER	358	34	6	99.17 ± 00.53	98.67 ± 01.13	93.60	94.00	93.92	7	31	11	15	11
ECO	336	7	8	87.86 ± 02.91	88.04 ± 03.37	76.78	77.47	81.39	7	3	10	9	11
GLA	214	10	6	97.50 ± 01.84	94.72 ± 02.97	54.05	56.62	53.37	7	63	11	20	9
OPT	5620	62	10	**99.07 ± 00.22**	96.95 ± 00.26	90.97	84.85	93.73	7	3	58	120	31
PAG	5473	10	5	97.37 ± 00.29	96.59 ± 00.22	95.70	96.71	94.65	7	3	16	23	29
PEN	10992	16	10	**99.54 ± 00.05**	95.53 ± 00.46	96.64	92.95	96.60	7	3	54	169	66
SEG	2310	18	7	**98.31 ± 00.37**	95.35 ± 00.77	93.99	94.48	91.96	7	7	22	41	33
YEA	1484	8	10	61.74 ± 00.73	60.00 ± 01.37	55.82	54.61	56.66	3	7	34	24	22

表6 100,000 次迭代下二分类的训练时间（分钟）

	N	D_x	GPHME 时间	BHME 时间
BRE	569	30	13.2187	52.3620
GER	1000	24	28.2990	17.5830
MAG	19020	10	32.0870	59.9671
MUS	6598	167	14.7300	10.0910
PIM	768	8	17.2795	55.8085
RIN	7400	20	27.1585	15.3693
SPA	4601	57	16.1985	10.5477
TWO	7400	20	34.2668	10.6366

表7 100,000 次迭代下多分类的训练时间（分钟）

	N	D_x	D_y	GPHME 时间	BHME 时间
BAL	625	4	3	16.2106	17.6489
CMC	1473	9	3	14.0640	10.9474
DER	358	34	6	28.6557	59.9738
ECO	336	7	8	32.4328	12.0421
GLA	214	10	6	21.6130	59.8710
OPT	5620	62	10	27.7151	10.0785
PAG	5473	10	5	19.7589	11.2346
PEN	10992	16	10	25.1264	10.2056
SEG	2310	18	7	26.9447	15.3570
YEA	1484	8	10	13.0739	15.3600

表 8 大规模数据集的准确率

	准确率		MNLL	
	GPHME	DGP	GPHME	DGP
MNIST8M	99.30%	99.14%	0.0372	0.0454
航空公司	71.79%	71.73%	0.5521	0.5510

参考文献

[1] M. I. Jordan 和 R. A. Jacobs，“分层专家混合与期望最大化算法”，Neural Computation，第 6 卷，第 2 期，第 181–214 页，1994年。[2] C. M. Bishop 和 M. Svens´en，“贝叶斯分层专家混合”，arXiv 预印本 arXiv:1212.2447，2012年。[3] A. Agrawal 和 J. Domke，“简单分层模型的摊销变分推断”，载于《神经信息处理系统进展》，A. Beygelzimer、Y. Dauphin、P. Liang 和 J. W. Vaughan 编，2021年。[4] W.-Y. Loh，“分类与回归树”，Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery，第 1 卷，第 1 期，第 14–23 页，2011年。[5] J. R. Quinlan，《C4. 5：机器学习程序》。Elsevier，2014年。[6] S. K. Murthy、S. Kasif 和 S. Salzberg，“一种用于归纳斜决策树的系统”，人工智能研究杂志，第 2 卷，第 1–32 页，1994年。[7] O. T. Yildiz 和 E. Alpaydin，“线性判别树”，国际模式识别与人工智能杂志，第 19 卷，第 03 期，第 323–353 页，2005年。[8] H. Guo 和 S. B. Gelfand，“利用神经网络特征提取的分类树”，载于《1992 IEEE计算机学会计算机视觉与模式识别会议论文集》。IEEE计算机学会，1992年，第 183–184 页。[9] C. Yildiz 和 E. Alpaydin，“全变量决策树”，IEEE神经网络汇刊，第 12 卷，1546，2001年 6，1539第 –期，第 。[10] O. Irsoy 、 O. T. Yıldız 和 E. Alpaydın，“ 软决策树 ”，载于《第 21 届国际模式识别会议（ICPR2012）论文集》。IEEE，2012年，第 1819–1822 页。

[11] J. Yang 和 J. Fan，“基于特征的分类的高斯软决策树”，载于亚太知识发现与数据挖掘会议。Springer，2021年，第143–155页。[12] N. Frosst 和 G. Hinton，“将神经网络提炼为软决策树”，arXiv 预印本 arXiv:1711.09784，2017年。[13] J. Q. Shi、R. Murray-Smith 和 D. M. Titterington，“用于回归的分层高斯过程混合模型”，Statistics and Computing，第15卷，第1期，第31–41页，2005年。[14] J. W. Ng 和 M. P. Deisenroth，“用于大规模高斯过程回归的分层专家混合模型”，arXiv 预印本 arXiv:1412.3078，2014年。[15] N. D. Lawrence 和 A. J. Moore，“ 分层高斯过程潜变量模型 ”，载于第 24 届国际机器学习会议论文集，2007年，第481–488页。[16] J. Lee、Y. Bahri、R. Novak、S. S. Schoenholz、J. Pennington 和 J. Sohl-Dickstein，“深度神经网络作为高斯过程”，arXiv 预印本 arXiv:1711.00165，2017年。[17] A. G. Wilson 和 P. Izmailov，“贝叶斯深度学习与泛化的概率视角”，神经信息处理系统进展，第33卷，第4697–4708页，2020年。[18] V. Dutordoir、J. Hensman、M. van der Wilk、C. H. Ek、Z. Ghahramani 和 N. Durrande，“深度神经网络作为深度高斯过程的点估计”，神经信息处理系统进展，第34卷，2021年。[19] G. Pleiss 和 J. P. Cunningham，“神经网络大宽度的局限性：深度高斯过程视角”，神经信息处理系统进展，第34卷，2021年。[20] R. M. Neal，《神经网络的贝叶斯学习》。Springer Science & Business Media，2012年，第118卷。[21] J. Quinonero-Candela 和 C. E. Rasmussen，“稀疏近似高斯过程回归的统一视角”，机器学习研究杂志，第6卷，第1939–1959页，2005年。[22] A. Rahimi 和 B. Recht，“面向大规模核机器的随机特征”，神经信息处理系统进展，第20卷，2007年。[23] A. Solin 和 S. S¨arkk¨a，“用于降秩高斯过程回归的希尔伯特空间方法”，Statistics and Computing，第30卷，第2期，第419–446页，2020年。[24] Y. Cho 和 L. Saul，“深度学习的核方法”，神经信息处理系统进展，第22卷，2009年。[25] Y. Gal 和 R. Turner，“通过表示频率输入中的不确定性改进高斯过程稀疏谱近似”，载于国际机器学习会议。PMLR，2015年，第655–664页。[26] R. B. Gramacy 和 H. K. H. Lee，“贝叶斯树状高斯过程模型及其在计算机建模中的应用”，美国统计学会会刊，第103卷，1130，2008年 483，1119 第 – 。[27] M. Maia、K. Murphy 和 A. C. Parnell，“GP-BART：一种采用高斯过程的新型贝叶斯加性回归树方法”，Computational Statistics & Data Analysis，第190卷，第107858页，2024年。[28] I. Achituve、A. Navon、Y. Yemini、G. Chechik 和 E. Fetaya，“GP-tree：用于少样本增量学习的高斯过程分类器”，载于国际机器学习会议。PMLR，2021年，第54–65页。[29] K. Cutajar、E. V. Bonilla、P. Michiardi 和 M. Filippone，“深度高斯过程的随机特征展开”，载于国际机器学习会议。PMLR，2017年，第884–893页。[30] P. Simard、D. Steinkraus 和 J. Platt，“卷积神经网络应用于视觉文档分析的最佳实践”，载于第七届国际文档分析与识别会议论文集。IEEE，2003年，第958–963页。[31] B. Sch¨olkopf，“支持向量学习”，博士论文，Oldenburg M¨unchen，德国，1997年。[32] J. Hensman、A. Matthews 和 Z. Ghahramani，“可扩展的变分高斯过程分类”，载于人工智能与统计学会议。PMLR，2015年，第351–360页。[33] K. Krauth、E. V. Bonilla、K. Cutajar 和 M. Filippone，“AutoGP：探索高斯过程模型的能力与局限性”，arXiv 预印本 arXiv:1610.05392，2016。[34] O. Irsoy、O. T. Yildiz 和 E. Alpaydin，“萌芽树”，载于 2014 22 2014，第 国际模式识别会议。IEEE，第3582–3587页。

[35]A. G. Wilson、Z. Hu、R. R. Salakhutdinov 和 E. P. Xing，《随机变分深度核学习》，《神经信息处理系统进展》，第 29 卷，2016 年。