

一种基于概率关联的局部高斯过程回归算法

肖文鑫¹ 张文文²

摘要 在针对控制和机器人的机器学习任务中, 高斯过程回归是一种常用方法, 具有无参数学习技术的优点. 然而, 它在面对大量训练数据时存在计算量大的缺点, 因此并不适用于实时更新模型的情况. 为了减少这种计算量, 使模型能够通过实时产生的大量数据不断更新, 本文提出了一种基于概率关联的局部高斯过程回归算法. 与其他局部回归模型相比, 该算法通过对多维局部空间模型边界的平滑处理, 使用紧凑支持的概率分布来划分局部模型中的数据, 得到了更好的预测精度. 另外, 还对更新预测矢量的计算方法进行了改进, 并使用 k-d 树最近邻搜索减少数据分配和预测的时间. 实验证明, 该算法在保持全局高斯过程回归预测精度的同时, 显著提升了计算效率, 并且预测精度远高于其他局部高斯过程回归模型. 该模型能够快速更新和预测, 满足工程中的在线学习的需求.

关键词 机器学习, 概率模型, 大数据量, 实时更新

引用格式 肖文鑫, 张文文. 一种基于概率关联的局部高斯过程回归算法. 自动化学报, 2022, 48(8): 1940–1949

DOI 10.16383/j.aas.c190445

A Local Gaussian Regression Algorithm Based on Probability Association

XIAO Wen-Xin¹ ZHANG Wen-Wen²

Abstract Gaussian regression is a common method in machine learning tasks for control and robotics, with the advantage of being a parametric learning technique. However, it has the disadvantage of being computationally intensive when faced with a large amount of training data, and thus is not suitable for the case of updating the model in real time. In order to reduce this amount of computation and realize the continuous updating of the model using a large amount of data generated in real time, this paper proposes a local regression algorithm based on probability correlation. Compared with other local regression models, the algorithm uses the tightly supported probability distribution to divide the data in the local model by smoothing the boundary of the multi-dimensional local space model and obtains better prediction accuracy. In addition, the calculation method of updating the prediction vector is improved, and the k-d tree nearest neighbor search is used to reduce the time of data allocation and prediction. Experiments show that the proposed algorithm improves the computational efficiency while maintaining the global regression prediction accuracy, and the prediction accuracy is much higher than other local regression models.

Key words Machine learning, probabilistic models, large data volumes, real-time update

Citation Xiao Wen-Xin, Zhang Wen-Wen. A local Gaussian regression algorithm based on probability association. *Acta Automatica Sinica*, 2022, 48(8): 1940–1949

高斯过程回归 (Gaussian process regression, GPR) 广泛应用于机器人、控制系统和航空航天飞行器等领域, 如机器人的逆动力学模型估计^[1-3]. 它是仅利用简单的线性代数处理非线性模型的有效工具, 提供了一种简单但有效的方法来表示数据的先验分布, 其特点是使用较少的参数得到准确的学习结果^[1]. 作为应用范围广泛的非参数回归, 它在样本外预测测试中脱颖而出, 已成为机器学习的重要手

段^[4]. 然而, 随着现实世界中训练集和测试集的增长, 高斯过程回归模型在面对大数据量时暴露了缺点. 当数据量 n 很大时, 标准高斯过程模型的立方复杂度 $O(n^3)$ 会导致不可接受的计算量, 使其不适用于高维系统或在线学习^[5].

降低这种复杂度的解决方案主要分为两种类型, 其中一种方法是稀疏高斯过程回归和高斯回归混合模型^[6-7]. 另一种是将数据分割并使用多个局部模型代替全局高斯过程模型, 这种方法称为局部高斯过程回归 (Local GPR, LGPR). 在稀疏高斯过程中, 按照设定的规则减少输入点的数量, 以降低计算复杂度^[8]. 局部高斯过程回归用多个局部子模型替换整体高斯过程模型, 由于每个局部模型具有较少数量的数据点并且被单独训练和更新, 因此总体

收稿日期 2019-06-17 录用日期 2019-10-16

Manuscript received June 17, 2019; accepted October 16, 2019

本文责任编辑 贺威

Recommended by Associate Editor HE Wei

1. 北京大学计算机学院 北京 100871 2. 同济大学电子与信息工程学院 上海 201804

1. School of Computer Science, Peking University, Beijing 100871 2. College of Electronics and Information Engineering, Tongji University, Shanghai 201804

计算成本降低^[4]. 对于局部高斯过程回归, 最关键的是如何划分数据并将其分配给不同的模型.

根据最近的文献, 夏嘉欣等^[9]提出一种基于噪声输入的稀疏高斯算法, 并将其应用于解决人体姿态估计问题. 乔少杰等^[10]利用高斯过程回归预测移动对象最可能运动轨迹. 王传云等^[11]对图像局部区域建立高斯混合模型解决图像的背景模型构建问题. Da 等^[12]研究了一种基于单任务聚合方案的局部高斯过程回归模型. Liu 等^[13]提出了一种基于流形学习方法的局部高斯过程相似度维持算法. Binois 等^[14]使用基于似然的高斯过程回归方法, 将异方差性问题表征为单目标函数, 使用完整的闭合形式导数, 实现了基于库的优化. 任志刚等^[15]利用加权的优秀样本预估高斯均值改进了传统高斯分布估计算法. Sarkar 等^[16]使用高斯过程回归构建了一种鲁棒算法, 能够处理不确定性和数据中的噪声, 并验证了相比经典谐波分析方法的优势. 文献 [17] 中的工作通过测量距离来选择合适的局部模型并最终计算加权预测, 但该方法在每次预测时仍然考虑所有模型, 因此不完全遵循局部支持的概念.

为了进一步降低计算量, 本文拟提出一种基于边界平滑过渡的局部高斯过程回归模型, 即训练集中的数据不是确定性地插入到单个局部模型中, 而是按概率分布分配给若干相邻的局部模型. 这意味着数据点属于某一模型的概率分布在特定区域内是连续的, 而在超出该区域的概率分布则为零. 这种分布将每个模型限制在一个定义清晰的区域, 通过仅考虑实际相关且可能对所考虑的数据点有贡献的模型来提高计算效率方面的性能, 也就是说对于特定的数据点只需要考虑少量数据点所属范围内的局部模型. 另外, 模型更新过程中使用矩阵块的求逆方法进行矩阵求逆, 大大减少了计算量. 为了实现更新与预测阶段局部模型的快速检索, 本文还引入并利用了来自计算几何领域的 k-d 树最近邻搜索算法^[18].

1 局部高斯过程回归模型

1.1 高斯过程回归

高斯过程回归是机器人和控制理论领域中用于监督机器学习的常用技术, 它的优点是通过少量的超参数实现高预测精度. 但是, 当数据点的总数 n 过大时, 高计算复杂度 $O(n^3)$ 将带来严重问题. 对于由 n 个数据点组成的数据集 $\{X, \mathbf{y}\}$, 每个数据点由 d 维输入数据 x 和一维输出数据 y 构成. 高斯过程回归假设输出数据由函数 $f(x)$ 和附加噪声 ϵ 组成

$$y_i = f(x_i) + \epsilon_i, \quad i \in \{1, 2, \dots, n\} \quad (1)$$

其中, y_i 是输出集 \mathbf{y} 中的第 i 点, x_i 是输入集 X 中的第 i 点, ϵ_i 是均值为零、方差为 σ_n^2 的正态分布噪声, n 是数据点的数量. 输出集满足下面的正态分布^[19]

$$\mathbf{y} \sim \mathcal{N}(0, K(X, X) + \sigma_n^2 I) \quad (2)$$

其中, $K(X, X)$ 是通过所有训练点计算得到的协方差矩阵, I 表示单位矩阵. 输出训练集 \mathbf{y} 和对问点 \mathbf{x}^* 的预测输出 $f(\mathbf{x}^*)$ 的联合分布由下式给出

$$\begin{bmatrix} \mathbf{y} \\ f(\mathbf{x}^*) \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, \mathbf{x}^*) \\ K^T(X, \mathbf{x}^*) & k(\mathbf{x}^*, \mathbf{x}^*) \end{bmatrix}\right) \quad (3)$$

$f(\mathbf{x}^*)$ 的条件分布服从一个新的正态分布

$$f(\mathbf{x}^* | \mathbf{y}, X) \sim \mathcal{N}(\mu(\mathbf{x}^*), \sigma(\mathbf{x}^*)) \quad (4)$$

其中, $f(\mathbf{x}^*)$ 的均值和方差的表达式为

$$\mu(\mathbf{x}^*) = K(\mathbf{x}^*, X) (K(X, X) + \sigma_n^2 I)^{-1} \mathbf{y} \quad (5)$$

$$\sigma(\mathbf{x}^*) = K(\mathbf{x}^*, \mathbf{x}^*) - K(\mathbf{x}^*, X) \times (K(X, X) + \sigma_n^2 I)^{-1} K(X, \mathbf{x}^*) \quad (6)$$

其中, $(K(X, X) + \sigma_n^2 I)^{-1} \mathbf{y}$ 项可用称为预测矢量的 $\boldsymbol{\alpha}$ 表示.

因此, 对问点 \mathbf{x}^* 的预测值可以通过下面表达式计算

$$y^* = K(\mathbf{x}^*, X) \boldsymbol{\alpha} \quad (7)$$

对于内核函数 $k(x_p, x_q)$ 的选择, 本文采用充分灵活的平方指数协方差函数, 其定义为

$$k(\mathbf{x}_p, \mathbf{x}_q) = \sigma_f^2 \exp\left[-\frac{1}{2} \sum_{i=1}^d \left(\frac{x_{p,i} - x_{q,i}}{l}\right)^2\right] \quad (8)$$

平方指数协方差函数只有两个自由参数, 即信号方差 $\sigma_f^2 \in \mathbf{R}_+$ 和长度标度 l . 加上前文提到的噪声的方差 σ_n , 这些参数称为高斯过程的超参数 $\boldsymbol{\theta} = [\sigma_n, \sigma_f, l]$. 本文通过最大化对数边际似然的方法来求得合适的超参数. 边际似然可由以下积分表示为

$$p(\mathbf{y} | X) = \int p(\mathbf{y} | \mathbf{f}, X) p(\mathbf{f} | X) d\mathbf{f} \quad (9)$$

使用文献 [20] 中提出的共轭梯度算法最大化对数边际似然

$$\begin{aligned} \log p(\mathbf{y} | X, \boldsymbol{\theta}) &= -\frac{1}{2} \mathbf{y}^T C^{-1} \mathbf{y} - \\ &\frac{1}{2} \log \det(C) - \frac{N}{2} \log 2\pi \end{aligned} \quad (10)$$

其中, $C = K(X, X) + \sigma_n^2 I$. 由以下对数边际似然导数的表达式可以推导出对数边际似然取最大值时的

超参数为

$$\frac{\partial}{\partial \theta} \log p(\mathbf{y}|X, \theta) = -\frac{1}{2} \mathbf{y}^T \mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \theta} \mathbf{C}^{-1} \mathbf{y} - \frac{1}{2} \log \det(\mathbf{C}) - \frac{1}{2} \text{tr} \left(\mathbf{C}^{-1} \frac{\partial \mathbf{C}}{\partial \theta} \right) \quad (11)$$

1.2 局部高斯过程回归

由于高斯过程回归在训练点数量大时表现不佳, 为了在保持高斯过程回归的优势的同时降低计算复杂度, 提出了局部高斯过程回归. 局部高斯过程回归可以实现模型的快速更新, 因此更适合使用在需要通过实时产生的数据不断更新模型的领域. 局部高斯过程回归方法背后的基本思想是引入多个回归模型并减少每个模型的数据点数量, 从而降低计算复杂度. 该方法将数据点分配到多个局部模型并单独训练^[1]. 对应于平方指数协方差函数, 点与模型中心之间的距离通过如下距离测量来计算, 以将数据点分配给最近的局部模型

$$w_i = \exp \left(-\frac{1}{2} \frac{(\mathbf{x} - \mathbf{c}_i)^T (\mathbf{x} - \mathbf{c}_i)}{l^2} \right) \quad (12)$$

其中, \mathbf{c}_i 表示的是第 i 个模型的中心, w_i 是模型 i 和给定 \mathbf{x} 之间的距离. 因此, 为了正确分配新数据点, 这个方法必须为每个数据点计算 M 次距离, 其中 M 是局部模型的总数. 基于此度量, 选择具有最小距离的模型, 并且将点添加到该最近的模型. 一旦选择了模型, 该方法后续步骤与第 1.1 节中的常规高斯过程回归完全相同. 如果没有足够接近新数据点的模型, 则会创建一个仅包含此新点的新模型. 亦即如果新点到所有中心的距离大于某个阈值, 则创建一个新模型, 其中心是新数据点.

式 (12) 中的距离度量也用于加权预测. 使用式 (7) 对第 i 个局部模型的预测值 y_i^* 进行计算, 然后通过对这些模型的加权预测求和来获得全局预测值

$$\mathbf{y}^* = \frac{\sum_{i=1}^M w_i y_i^*}{\sum_{i=1}^M w_i} \quad (13)$$

由于每个局部模型中的数据点的数量比高斯过程回归中的低, 用于更新模型的计算工作量得以减少, 并且与高斯过程回归相比可以实现整体性能改进. 在第 2.1 节, 本文提出的算法对以上方法进行了改进, 采用概率数据关联将数据点在其最近的 2^d 个局部模型中进行分配.

1.3 高效 k 近邻查询树

为了实现快速搜索算法, 本文使用 k -d 树来存储数据. 利用 k -d 树进行局部模型检索, 可以有效提高模型更新与对未知数据点预测的速度. k -d 树是二叉搜索树 (Binary search tree, BST) 的多维形式, 而二叉搜索树是用于快速可靠查询的合适方法^[18]. 文献 [21] 中提出了这个 k -d 树的优化版本.

1.3.1 k -d 树的创建

对于树的第零层, 首先在第 1 个维度分割数据, 并使用此维度中数据的中值作为分离的边界. 此步骤将数据集分为两部分: 一部分是由小于或等于中值的数据组成的树的左侧; 另一部分是剩余数据构成的树的右侧. 在每个子树上重复此过程, 直到只剩下包含一个点的节点, 将其作为树的子节点. 由于在每个节点中重复划分并将剩余的边处理为新的子树, 因此这是一个递归定义. 分割的维度是根据树中当前节点的级别来决定的, 即在层级 i 上, 选择维度 $i+1$ 进行分割. 如果迭代遍历了所有维度, 则在树的下一层过程将再次从第 1 个维度开始分割, 直到只剩下具有一个数据点的子集表示树的子节点.

1.3.2 k -d 树最近邻搜索

k -d 树最近邻搜索是一种快速且可靠的搜索算法, 其基于预定义的相似性或距离度量来判断距查询点最近的 k 个局部模型. 对于如何找到最佳匹配和 k 个最近邻项存在多种方法^[22-23], 这项工作将搜索算法制定为递归策略, 其中算法的输入是每次递归对应的节点和其子节点. 首先, 包含所有数据的根节点作为首次的输入传递到算法. 当前最近邻项的候选项被存储在优先队列中, 其中查询点和近邻项之间的相似性是判断的依据. 接下来会将每个节点与优先队列中的条目进行比较, 如果该节点比队列中的一个条目更接近查询点, 则相应地更新优先队列使该节点成为新的候选项. 如果节点不是终端, 即不是叶子节点, 则查询点所在的子树再次递归地传递给算法. 如果节点是叶子节点, 则此子树上的递归结束. k -d 树最近邻搜索的计算复杂度是 $O(k \times n \times \log n)$, 其中 k 是数据的维度, n 是数据点的个数.

2 基于概率关联进行模型选择的局部高斯过程回归

本研究提出一种无硬边界局部模型概率关联方法, 这种方法确保了在仅有局部支持的条件下实现模型之间的平滑过渡. 模型建立阶段, 通过定义模型相邻部分的概率分布, 数据点根据其坐标分配

到局部模型中. 对未知数据预测阶段, 由距离查询点最近的若干模型计算加权平均值, 从而求和得到全局预测值. 因此, 该算法包括 3 个阶段: 1) 初始模型的建立; 2) 局部模型的更新; 3) 测试集的加权预测. 第 2.1 节将讨论局部模型的概率型数据关联, 第 2.2 节和第 2.3 节描述了模型的建立和模型的更新, 第 2.4 节提出了对未知数据的预测方法.

2.1 边界约束的局部高斯过程回归模型

紧致支持的边界约束局部模型是本在线学习模型的基本组成部分, 本节介绍如何构建本地模型.

根据文献 [1, 17] 局部高斯过程回归方法通常使用模型内数据点的平均值作为每个模型的中心. 因此, 对于每个新输入的数据点, 这些中心都需要更新, 即局部模型需要持续地改变. 为了减少每次更新造成的计算量, 本文提出一种固定的模型网络, 同时为了实现局部模型与数据之间的概率关联, 我们提出了一种激活函数. 激活函数 $\phi(c_i|\mathbf{x})$ 表示数据点 \mathbf{x} 属于局部模型 c_i 的概率, 满足概率分布的条件. 距数据点最近的 2^d 个局部模型对 \mathbf{x} 的激活函数可以组成一个离散概率密度函数 $f(c)$, 其中, d 是输入数据点 \mathbf{x} 的维度, $\mathbf{c} = c_1, \dots, c_i, \dots, c_{2^d}$, c_i 是局部模型 i 的中心, 本文以模型中心代指具体的模型. 最后, 根据 $f(c)$ 中的概率分布产生一个伪随机数, 以决定数据点 (\mathbf{x}, y) 分配到哪个局部模型中.

所有局部模型组成一个覆盖数据集空间的模型网格, 在维度为一维时, 该网格由每个中心之间的距离 w 定义. 当维度 k 增加时, 在所有维度中应用这种设置来延伸网格, 从而生成由各局部模型组成 k 维立方体.

相邻的局部模型在共享边界处做出不同的预测, 导致边界处的预测是不连续的^[24]. 实际上, 对于几个模型的交汇处的点, 属于每个模型的概率可能都大于 0, 这意味着它不是绝对属于最近的模型. 然而, 当局部模型被设计为具有硬边界时, 预测值仅由最近的模型确定, 这可能导致大的误差. 为了实现相邻局部模型之间的平滑过渡, 在创建局部高斯过程回归模型时, 本文中使用紧致支持的边界约束来构建局部模型. 这里采用激活函数定义模型的边界约束, 以表示某一坐标的数据点属于这个模型的概率. 在模型中心位置, 定义只有这个模型是“激活”的, 这意味着在这个位置模型 c_i 的激活函数 $\phi(c_i|\mathbf{x})$ 等于 1, 其余模型的激活函数等于 0, 位于这个位置的数据点只可能属于这个模型. 通过这样的设置, 局部模型与其相邻的各模型共享边界区域. 由于本算法面向的是工程中在线更新的场景, 主要

处理的是多维数据, 因此激活函数应当尽可能考虑对称的形式, 以避免多维空间中模型构造过于复杂. 基于通过简洁的结构实现更好的性能的理念, 将共享区域的激活函数定义为范围从 0 到 1 的斜坡函数. 这样, 所有局部模型使用相同形式的激活函数. 如图 1 所示, 激活函数可表示为

$$\phi(c_i|\mathbf{x}) = 1 - \frac{|\mathbf{x} - c_i|}{w} \quad (14)$$

其中, \mathbf{x} 是数据点在当前维度的值, c_i 是局部模型的中心在当前维度的坐标, w 是相邻局部模型的间距. 由图中可以看出, 每个局部模型在当前维度的宽度为 $2w$. 对于处于相邻模型中心 c_i 和 c_{i+1} 之间的 \mathbf{x} , 满足

$$\phi(c_i|\mathbf{x}) + \phi(c_{i+1}|\mathbf{x}) = 1 \quad (15)$$

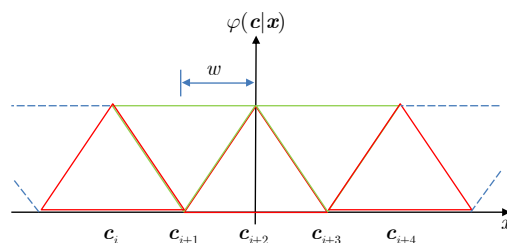


图 1 一维局部模型激活函数示意图

Fig. 1 Schematic diagram of one-dimensional local model activation function

由此该激活函数实现了数据点与邻近局部模型的概率关联.

图 1 为一维情况下局部模型的初始化示意图. 基于这种一维方法, 本文开发了具有更高维度的数据的解决方案. 对于模型布局 and 激活功能, 基本原理保持不变. 但是, 每个数据点可能被分配进的模型最多为 2^d 个. 将一维网格布局自然延伸至 d 维, 得到多维模型网络.

图 2 展通过在二维空间内构建 c_1, c_2, c_3, c_4 为中心的 4 个局部模型, 展示了训练集数据为二维时的局部模型布局 and 可能激活的模型的情况. 示意图中的 c_1, c_2, c_3, c_4 分别为 4 个局部模型的中心. 图中的区域 A 由于不与其他局部模型相交, 此区域内的点只可能属于模型 c_1 , 区域 B 与一维时的情况相同, 最多只有两个模型处于激活状态, 而这些激活中心 c_1 和 c_2 共享相同的第 2 维分量. 对于区域 A 中的点, 可能的激活局部模型仅是 c_1 模型; 对于区域 B, 可能的激活模型是 c_1 和 c_2 模型. C 区域则有 4 个可能激活的模型. 另外, 该图还显示了各局部模型在二维立方体的等距网格中的位置.

对于多维数据, 使用输入数据点在每个维度上

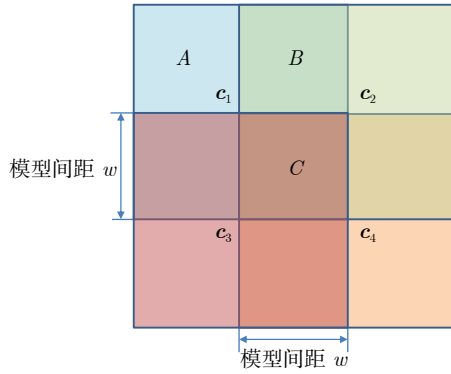


图 2 二维局部模型分布示意图

Fig. 2 Schematic diagram of two-dimensional local model distribution

的最大值和最小值来计算各局部模型的中心. 局部模型的数量和中心可表示为

$$num_i = \text{ceil} \left(\frac{f_{\max_i} - f_{\min_i}}{w_{d_i}} \right) + 1 \quad (16)$$

$$c_{i,n} = f_{\min_i} + n \times w_{d_i} \quad (17)$$

其中, num_i 是局部模型在维度 i 上的个数, f_{\max_i} 和 f_{\min_i} 分别是输入数据在维度 i 上的最大值和最小值, w_{d_i} 是相邻模型在维度 i 上的间距, $c_{i,n}$ 表示的是第 i 维第 n 个模型中心的位置. 维度 i 上模型中心的最小值设置为 f_{\min_i} , 以保证所有数据点在各维度属于各局部模型的概率和始终为 1. 因此, 图 1 中的最外侧的两个局部模型均只利用了靠内的一半. 扩展到图 2 的情况, 数据点范围是集中在 C 区域中的, 其他区域由于处在整个模型的边缘部分, 未加利用. 当模型网格扩大时, 始终未使用整个模型的边缘区域. 计算出局部模型网络中所有模型中心的坐标后, 用这些坐标值建立 k - d 树, 每个模型都用一个叶节点表示. 通过这个数据结构, 可以使用文献 [22] 中的快速搜索算法来执行有效的最近邻居查询, 以便后续更新模型和预测未知输出时搜索模型时使用.

初步确定模型的位置后, 根据式 (14) 的激活函数将训练集中的所有数据点分配到各个局部模型中. 在每个维度上模型中心之间的距离都是 w , 使得局部模型的范围覆盖边长为 w 的 d 维立方体. 这些立方体的中心即为局部模型的中心. 在整个模型的非边缘区域, 每个位置均由 2^d 个立方体覆盖, 也就是说可以有 2^d 个激活模型. 为了得到 d 维情况下的概率分布, 首先使用式 (14) 在每个维度上计算局部模型对应的激活函数 $\phi_t(c_i|\mathbf{x})$, 它们仅取决于该特定维度中的位置, 使得它们符合独立分布的概率. 然后通过这些一维值的乘积来计算整体激活函数 $\phi(c_i|\mathbf{x})$, 局部模型 c_i 的全局激活函数由其所有子概

率的乘积得到, 即

$$\phi(c_i|\mathbf{x}) = \prod_{t=1}^d \phi_t(c_i|\mathbf{x}) = \prod_{t=1}^d \phi_{d,i} \quad (18)$$

2.2 局部模型的初始化

本文提出的在线学习模型由初始化和更新阶段两个阶段建立, 本节解释了如何使用训练集构建初始局部模型.

为了初始化模型, 本研究使用了一种对训练集数据点进行聚类从而创建局部模型的方法. 首先, 通过训练集中的数据计算各个局部模型中心的坐标, 构建第 2.1 节中所述的模型网络. 然后将所有数据点分配到相应的局部模型中. 为了确定最终分配数据点的模型 c_i , 必须在具有非零概率的激活模型中确定一个模型. 如果直接选择最有可能的模型, 会在模型相邻处形成硬边界, 导致预测在这个区域的不连续性. 为了解决这个问题, 对于训练集中的数据点 (\mathbf{x}, y) , 通过第 1.3.2 节介绍的 k - d 树实现 k -最近邻搜索, 找到距离数据点最近的 2^d 个局部模型的编号. 通过式 (18) 计算数据点对应这些局部模型的激活函数 $\phi(c_i|\mathbf{x})$ 的值, 可得到至多 2^d 个非零概率, 构造由这些概率值组成的离散概率密度函数 $f(\mathbf{c})$, 其中的概率值分别对应一个局部模型. 根据各模型对应的概率从 $f(\mathbf{c})$ 中抽取随机数 i , 确定激活的局部模型 c_i , 将数据点分配到这个模型. 每个数据点仅分配给一个模型, 但与其他局部高斯过程回归方法不同, 这种关联是用概率方法完成的. 完成训练集中所有数据点的分配后, 通过第 2.1 节中的方法计算每个局部模型的预测矢量 α , 并创建相应的 LGPR 模型.

2.3 局部模型的更新

本文提出的紧致支持的边界约束局部模型最突出的特点是支持在线学习, 这意味着局部模型能够通过实时数据不断进行更新. 模型的在线更新可以应对一些现实世界的工程问题, 例如机械特性和机器人交互随时间的变化^[25]. 在更新阶段, 依据第 2.2 节中分配数据点的方法, 新输入点 (\mathbf{x}, y) 分配到一个局部模型, 需要对这个模型的参数进行更新, 供预测阶段调用. 对未知点的预测是通过式 (7) 计算得到, 其中协方差函数 $K(\mathbf{x}^*, X)$ 只能在查询点 \mathbf{x}^* 已知的情况下计算, 而预测矢量 $\alpha = (K(X, X) + \sigma_n^2 I)^{-1} \mathbf{y}$ 可以在模型更新阶段利用已知数据更新, 供预测阶段直接调用.

为了更新预测矢量, 首先, 将局部模型的数据集更新为

$$X_n = [X, \mathbf{x}], \quad \mathbf{y}_n = [\mathbf{y}, y] \quad (19)$$

其次, 预测矢量 $\boldsymbol{\alpha}$ 中更新后的协方差矩阵 $K_n(X_n, X_n) + \sigma_n^2 I$ 项可以表示为

$$K_n(X_n, X_n) + \sigma_n^2 I = \begin{bmatrix} K(X, X) + \sigma_n^2 I & K(X, \mathbf{x}) \\ K(\mathbf{x}, X) & k(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I \end{bmatrix} \quad (20)$$

其中, $K(X, X) + \sigma_n^2 I$ 是更新前的协方差矩阵, $K(X, \mathbf{x})$ 是由 X 和 \mathbf{x} 的协方差组成的矢量, $K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I$ 由 \mathbf{x} 的自协方差和模型的超参数 σ_n^2 相加得到. 使用式 (20) 对协方差矩阵求逆, 得到

$$(K_n(X_n, X_n) + \sigma_n^2 I)^{-1} = \begin{bmatrix} K^{-1}(I + \mathbf{b}\mathbf{b}^\top K^{-1} \frac{1}{\alpha}) & -K^{-1} \mathbf{b} \frac{1}{\alpha} \\ -\mathbf{b}^\top K^{-1} \frac{1}{\alpha} & \frac{1}{\alpha} \end{bmatrix} \quad (21)$$

其中

$$\mathbf{a} = K(\mathbf{x}, \mathbf{x}) + \sigma_n^2 I - K^\top(X, \mathbf{x})K^{-1} - K(X, \mathbf{x}) \quad (22)$$

$$\mathbf{b} = K(X, \mathbf{x}) \quad (23)$$

$$K^{-1} = (K(X, X) + \sigma_n^2 I)^{-1} \quad (24)$$

根据以上方法得到 $(K_n(X_n, X_n) + \sigma_n^2 I)^{-1}$ 和 \mathbf{y}_n 后, 可以通过下式计算更新的预测矢量, 即

$$\boldsymbol{\alpha}_n = (K_n(X_n, X_n) + \sigma_n^2 I)^{-1} \mathbf{y}_n \quad (25)$$

由式 (21) 可知, 每次在计算新的协方差矩阵的逆时, 都利用了它在更新前的值以减少计算量. 因此, 在模型的更新阶段需要更新并储存数据集、协方差矩阵的逆和预测矢量.

如果在使用新数据点更新模型阶段检测到数据点不在任何已有局部模型范围内, 即在储存局部模型的 k - d 树中没有搜索到足够接近数据点的模型, 则会创建一个仅包含此新点的新模型, 其中心是新数据点, 并补齐模型网格的其余部分. 最后使用第 1.1 节中的方法计算这个模型的超参数 $\boldsymbol{\theta}$ 和预测矢量 $\boldsymbol{\alpha}$, 并创建 LGPR 模型, 计算新建模型的中心坐标和间距参数.

2.4 局部模型对未知数据的预测

基于输入数据预测输出数据是本局部高斯过程回归模型的一个重要功能. 在多个模型处于激活状态的区域, 可以通过激活函数计算查询点属于各局部模型的概率, 再通过各模型的预测结果计算加权预测. 首先, 通过 k - d 树搜索查找到距查询点 \mathbf{x}^* 最近的 2^d 个局部模型, 使用激活函数 $\phi(\mathbf{c}_i|\mathbf{x})$ 算出各模型对应的概率, 这些概率值组成离散概率密度函数 $f(\mathbf{c})$. 基于这些概率值和这 2^d 个局部模型各自的

预测值, 可以计算得到全局加权预测值. 如第 1.1 节所述, 局部模型 \mathbf{c}_i 的预测遵循高斯分布, 其平均值为

$$\bar{y}_i^* = K(\mathbf{x}^*, X_i) \boldsymbol{\alpha}_i \quad (26)$$

其中, \mathbf{x}^* 是输入的查询点, X_i 是分配到模型 \mathbf{c}_i 中的输入数据点集, $\boldsymbol{\alpha}_i$ 是第 i 个局部模型的预测矢量.

通过每个局部模型对应的概率对其预测值进行加权求和, 计算对查询点 \mathbf{x}^* 的预测输出 y^* . 如前所述, 激活模型 \mathbf{c}_i 的概率 $P(\mathbf{c}_i|\mathbf{x})$ 通过激活函数 $\phi(\mathbf{c}_i|\mathbf{x})$ 表示, 可以导出 y^* 的期望值的表达式为

$$\bar{y}^* = \sum_{i=1}^{2^d} \phi(\mathbf{c}_i|\mathbf{x}) \bar{y}_i^* \quad (27)$$

因此, 最终的全局预测值表示为

$$y^* = \sum_{i=1}^{2^d} \phi(\mathbf{c}_i|\mathbf{x}) \bar{y}_i^* K(\mathbf{x}^*, X_i) \boldsymbol{\alpha}_i \quad (28)$$

3 实验结果及分析

本节通过实验, 在更新时间、预测时间和误差三个方面评估边界约束的局部高斯过程回归算法的性能. 第 3.1 节通过人工生成的低维数据, 将本算法与全局高斯过程回归方法和具有硬边界的局部高斯回归过程方法进行比较, 验证了低维数据时本算法的优势. 第 3.2 节使用来自 F16 飞机的真实多维工程数据进行实验并将本算法应用于机械手臂上, 分析了本算法在多维数据和大数据量的场景下, 实现在线学习的可行性.

3.1 低维数据实验

为了验证该基于边界约束的局部高斯过程回归算法的有效性, 本文通过模拟带有噪声的高斯过程对该算法进行测试, 并与全局高斯过程回归方法和具有硬边界的局部高斯回归过程方法进行比较^[17, 19]. 对于训练和测试, 数据集由正弦函数的组合产生, 超参数通过第 1.1 节中的方法计算. 本算法的性能取决于局部模型间距参数的选择, 本文首先评估它们对预测结果的影响. 通过下面正弦函数组合生成二维数据集

$$y(\mathbf{x}) = -\cos(0.05x_1) \sin(0.1x_2 - 1) - 0.5 \quad (29)$$

并为输出 y 附加 $\sigma_n = 0.1$ 的噪声. 首先, 将各维度的输入值都限定在 0 到 40 之间, 取 6 400 个数据点作为总的数据集. 从中随机选取 80% 作为训练集, 剩余部分用于测试集. 使用这组数据重复模拟边界约束的高斯局部模型 50 次, 得到了不同参数情况下的平均预测误差与运行时间, 如图 3 所示.

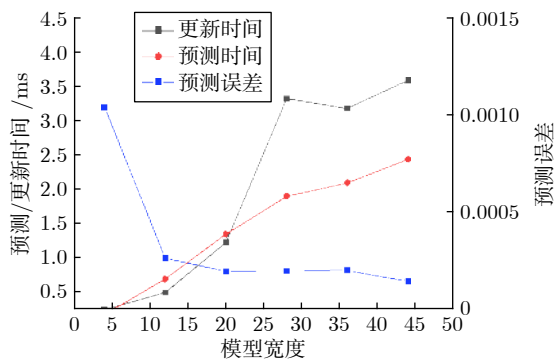


图 3 局部模型参数对边界约束模型性能的影响

Fig. 3 Influence of local model parameters on performance of boundary constraint model

图中展示了局部模型间距以 8 为间隔由 4 增加到 44 时, 用单个数据点更新模型和预测未知输出所用时间及预测误差值的变化情况. 小的局部模型宽度意味着总的模型数量多, 即每个局部模型分配的数据点较少, 对应着较大的误差和较短的模型更新和预测时间. 当局部模型宽度增加时, 每个局部模型分配的数据点也随之增加, 使得预测结果更为准确. 但同时由于求协方差逆矩阵的计算工作量随着更多数据点而上升, 消耗的时间也随模型宽度增加而延长. 因此, 模型设计过程中需要对预测精度和计算时间做出权衡. 由图中可见, 当模型宽度增加到 20 以后, 模型预测的准确度变化不大, 但更新和预测时间仍然随模型宽度增加而大幅增加, 因此 20 是这个数据集范围内较好的局部模型宽度参数.

为了直观地比较 3 种方法的预测结果, 本文使用以下等式生成一维输入与输出数据集

$$y(\mathbf{x}) = 0.5 \sin(\mathbf{x} - 1)^2 + 3 \sin(0.1\mathbf{x}) + 1 \quad (30)$$

图 4 展示了 3 种方法的预测情况, 其中本文提出的算法的预测误差为 0.03292, 全局回归方法的误差为 0.03239, 硬边界的局部回归方法的误差为 0.50499. 在这个数据集中, 本算法实现了与全局高斯过程回归同一量级的精度, 比硬边界局部高斯过程回归方法的精度高一个数量级.

为了进一步比较 3 种算法的性能, 本文使用式 (24) 生成的二维数据, 将 3 种方法的模型各重复 50 次, 分别得到它们的平均预测误差、单个数据点的更新和预测时间, 如表 1 所示. 其中, 边界约束的局部高斯过程模型的模型宽度参数设置为 20.

从表中可以看出, 全局高斯过程回归方法的预测精度最佳, 而本算法的预测误差与它处于同一数量级, 硬边界局部模型方法的预测误差约为全局高斯过程回归方法的 80 倍. 考虑到全局高斯过程回

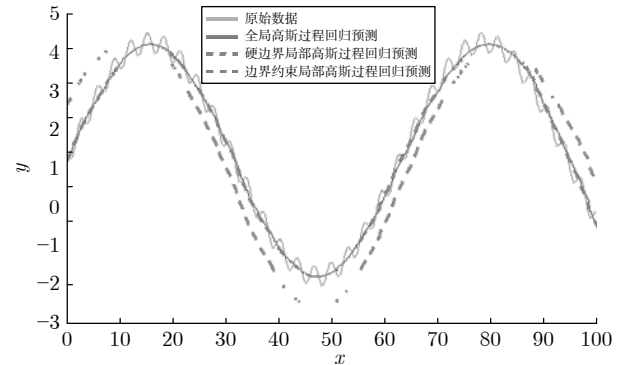


图 4 对一维测试集的预测结果

Fig. 4 Prediction results for one-dimensional test sets

表 1 3 种方法的性能对比

Table 1 Performance comparison of three methods

	全局 GPR	硬边界 LGPR	边界约束 LGPR
预测误差	1.281×10^{-4}	97.775×10^{-4}	1.953×10^{-4}
更新时间 (ms)	132.753	0.929	1.230
预测时间 (ms)	2.190	2.371	1.342

归使用全部数据构建单个模型, 本算法的精度稍低于它是可以接受的. 本算法通过与硬边界局部模型方法接近的局部模型数量 (各约 100 个), 实现了更好的预测精度. 比较更新模型的时间时, 全局高斯过程回归方法的弊端显现出来, 其使用单个数据点更新模型的时间为 132.753 ms, 是局部模型方法的 100 倍以上. 当数据量继续增大时, 计算量将迅速增大直至无法被接受, 而本算法和硬边界局部模型方法的模型更新时间处于同一数量级. 另外, 3 种方法对单个数据点的预测时间为同一数量级, 其中本文算法的预测时间最短, 为 1.342 ms. 综上所述, 本文提出的算法在保证高精度预测的同时, 能大量减少局部模型的更新时间.

3.2 多维工程数据实验

在机器人控制和航空航天等工程领域, 输入数据通常是多维的, 离线学习的数据集很难覆盖所有情景. 即时能够使用大量数据集提前进行离线学习, 计算量也会是巨大的. 针对这些工程场景, 本文提出的在线学习模型能够使用实时数据更新模型, 实现预测功能.

为了分析该模型处理真实多维工程数据时的性能, 本文采用葡萄牙波尔图大学公开的来自 F16 型号飞机升降电梯的数据集进行实验^[26]. F16 型号飞机升降电梯的数据集共包括 9 517 个数据点. 输入数据有 6 个属性: 爬升率、高度、转轮角速度、转轮角度、爬升加速度和爬升加速度的变化率, 这 6 个

属性的数值组成了 6 维输入数据. 输出是控制信号 Se , 它是一个处于 -0.014 到 0.013 间的值. 为了分析在线学习过程中, 边界约束回归模型的性能是如何随着数据量的增长变化的, 将 9 000 个数据点作为训练集, 以数据流的形式不断输入模型进行更新. 每增加 1 000 个数据点, 记录当前数据量对应的更新时间、预测时间和误差. 这 3 项是判断模型是否满足在线学习要求的关键指标, 下面对这 3 项指标进行分析.

首先, 用一个新数据点更新模型所需的时间 T_u 是衡量在线学习模型性能的重要指标, 只有模型能够随数据流快速更新, 才能实现在线学习. 图 5 展示了 T_u 随数据量增长的变化趋势.

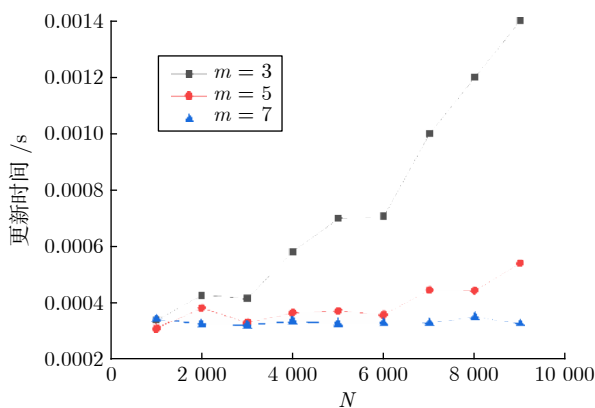


图 5 更新时间随数据量增长的变化趋势

Fig.5 The trend of update time as data increases

图 5 中 N 是用于训练模型的数据点的数量, m 是每个维度的局部模型数量, 通过 m 和每个维度输入数据的范围可以确定模型宽度. 可以看到, 当 $m=3$ 时, T_u 增长的趋势过快, 而 $m=5$ 时 T_u 由 $N=1000$ 到 $N=9000$ 只增加了 0.75 倍, $m=7$ 时 T_u 仅小幅度波动. 因此 $m=5$ 或 7 对于此数据集是更为合适的参数.

图 6 展示了对一个未知数据点进行预测所需的时间 T_p 随数据量增长的变化趋势. 同样显示了 $m=5$ 或 7 时预测时间增长平缓, 分别只增加了 0.75 倍和 0.28 倍, 相比 $m=3$ 是更合适的参数.

图 7 显示了平均预测误差随数据量增长的变化趋势. 该图的趋势表明, 随着训练集中数据点的增加, 模型的预测误差减少. 也就是说随着数据流的输入, 在线学习模型能够越来越精确.

综合图 5 ~ 7, 对于这一特定数据集 $m=5$ 是最合适的参数. 在 $m=5$ 时, 数据点数量 $N=9000$ 虽然是 $N=1000$ 时的 9 倍, 但它的更新时间 0.541 ms 只是 $N=1000$ 时的 0.309 ms 的 1.75 倍, 能够实现

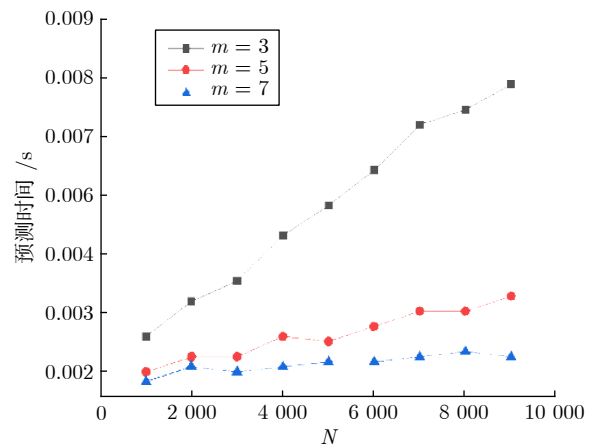


图 6 预测时间随数据量增长的变化趋势

Fig.6 The trend of prediction time as the data increases

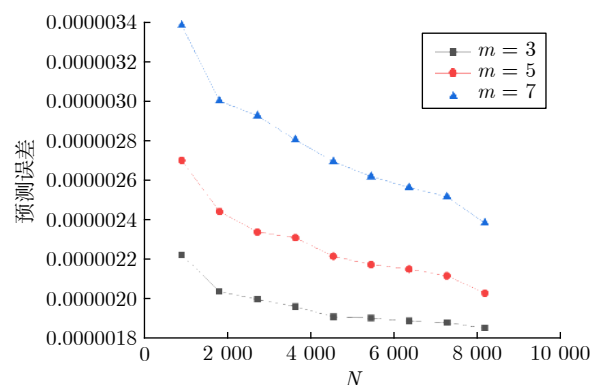


图 7 预测误差随数据量增长的变化趋势

Fig.7 The trend of prediction error as data increases

快速更新模型, 满足在线学习的要求. 预测时间由 $N=1000$ 时的 2 ms 增加到 $N=9000$ 时的 3.5 ms, 只增加了 0.75 倍, 同样满足在线学习的实时性要求. 本模型的预测误差由 $N=1000$ 时的 0.0000026989 降低到 $N=9000$ 时的 0.0000020318, 随着数据量的增加实现了更高的精度.

为了进一步验证本模型在机器人控制场景下的可用性, 本文将模型应用到具有两个自由度的机器人的机械手臂上进行实验, 测试其在机器人控制方面应用的可行性. 该机器人手臂由文献 [27] 中提出的基于 PD 反馈的计算扭矩控制方法进行控制. 我们为机器人手臂设置期望的二维平面活动轨迹, 通过模型的在线学习, 预测控制机械臂所需的关节扭矩, 并以此控制机器人手臂使得它能按照期望的活动轨迹运动. 本文通过下式为机器人手臂设定期望的活动轨迹

$$\theta_1 = \sin(t) \quad (31)$$

$$\theta_2 = 3 \cos(t) \quad (32)$$

其中, θ_1 和 θ_2 分别是两段机器人手臂与水平线的夹角, t 是从 0 开始计时的时间. 将 θ_1 和 θ_2 对 t 求导, 得到两段机器人手臂的角速度 ω_1 和 ω_2 . 由 θ_1 , θ_2 , ω_1 和 ω_2 组成四维输入矢量 $\mathbf{x} = [\theta_1 \ \theta_2 \ \omega_1 \ \omega_2]$. 我们的任务是使用这些输入数据通过本文模型预测机器人手臂的关节扭矩, 并使用该关节扭矩控制机械臂, 使得机械臂能够跟随预设轨迹, 同时将四维输入数据和对应的真实关节扭矩输入模型进行在线学习. 为了更清晰地展现控制效果, 本文截取了机械臂 10 s 内的运动轨迹参数, 共计 2 000 个预测点, 即每 0.005 s 预测一次, 对模型有实时性要求.

图 8 显示了预设和实际运动轨迹分别对应的参数随时间的变化情况, 其中, 4 条由星号绘制的粗曲线分别对应的是实际运动轨迹的角度和角速度, 4 条虚线分别对应的是预设运动轨迹的角度和角速度. 从图中可以看出, 在开始阶段, 实际运动轨迹和预设运动轨迹的曲线相差较大, 但随着模型的运行, 两者的曲线趋于重合.

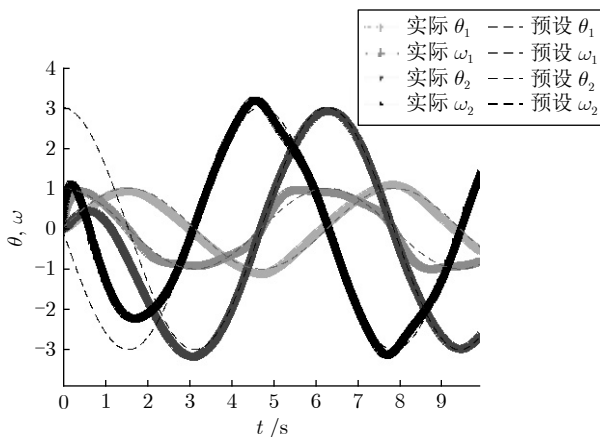


图 8 预设和实际运动轨迹的参数

Fig.8 Preset and actual motion trajectory parameters

为了更直观地展示图 8 中预设和实际运动轨迹间的误差变化情况, 使用轨迹误差矢量的范数 $\|\mathbf{x}_r - \mathbf{x}\|$ 表示实际运动轨迹和预设运动轨迹之间的误差

$$\text{error} = [(\theta_{r,1} - \theta_1)^2 + (\theta_{r,2} - \theta_2)^2 + (\omega_{r,1} - \omega_1)^2 + (\omega_{r,2} - \omega_2)^2]^{\frac{1}{2}} \quad (33)$$

其中, $\theta_{r,1}$, $\theta_{r,2}$, $\omega_{r,1}$, $\omega_{r,2}$ 是实际运动轨迹的参数, θ_1 , θ_2 , ω_1 , ω_2 是预设运动轨迹的参数. 由此得到图 9, 它显示了运动轨迹误差随着时间振荡下降的趋势. 图 8 和图 9 验证了本文模型在高实时性要求的场景

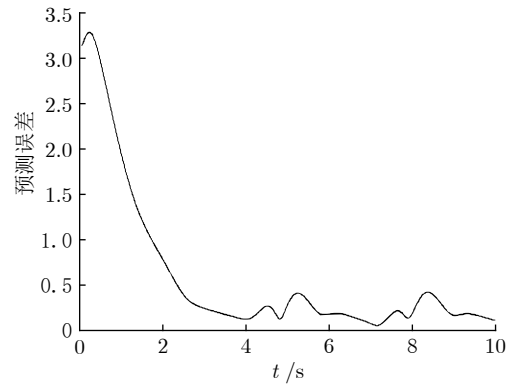


图 9 运动轨迹误差

Fig.9 Motion track error

下的可用性.

以上各实验从不同角度验证了本文模型能够适用于工程中的大数据量场景, 并且满足在线学习的实时性需求.

4 结束语

为了在保持全局高斯过程回归预测精度高等优点的情况下, 减小其立方复杂度带来的过大的计算量, 本文提出了一种基于边界约束的概率相关局部高斯过程回归模型. 本文在该模型中使用一种基于贝叶斯原理的数据关联方法, 定义了用于数据点的分配和预测的激活函数. 同时, 本文将局部模型放置在空间网格中, 实现空间重叠的局部模型设置, 提高了模型的精度. 在算法方面, 利用对计算过程的优化和 k-d 树最近邻搜索, 以及更新模型预测矢量时采用已储存数据求矩阵的逆, 降低了模型更新阶段的计算量. 评估结果表明, 与其他局部高斯过程回归方法相比, 本文方法可以显著缩短计算时间, 同时实现与全局高斯过程回归方法相似的预测精度, 能够更好地处理大量且持续更新的数据. 在真实工程场景下, 能够很好地完成在线学习功能. 该模型具有能够快速更新和预测的特点, 满足在线学习的需求, 对于机器人控制、航空航天等实时产生大量多维数据的领域, 具有一定的工程意义.

References

- Schneider M, Ertel W. Robot learning by demonstration with local Gaussian process regression. In: Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems. Taipei, China: IEEE, 2010. 255-260
- Hoefling F, Franosch T. Anomalous transport in the crowded world of biological cells. *Reports on Progress in Physics*, 2013, 76(4): 046602
- Qiu L, Yuan S F, Chang F K, Bao Q, Mei H F. Online updating Gaussian mixture model for aircraft wing spar damage evaluation under time-varying boundary condition. *Smart Materials and Structures*, 2014, 23(12): 125001

- 4 Gramacy R B, Apley D W. Local Gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 2015, **24**(2): 561–578
- 5 Bishop C M. *Pattern Recognition and Machine Learning*. Berlin: Springer-Verlag, 2006. 306–311
- 6 Candela J Q, Rasmussen C E. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 2005, **6**(12): 1939–1959
- 7 Rasmussen C E. The infinite Gaussian mixture model. *Advances in Neural Information Processing Systems*, 2000, **12**: 554–560
- 8 Plagemann C. Gaussian Processes for Flexible Robot Learning [Ph.D. dissertation], Albert Ludwig University of Freiburg, Germany, 2008.
- 9 Xia Jia-Xin, Chen Xi, Lin Jin-Xing, Li Wei-Peng, Wu Qi. Sparse Gaussian process with input noise for human pose estimation. *Acta Automatica Sinica*, 2019, **45**(4): 693–705 (夏嘉欣, 陈曦, 林金星, 李伟鹏, 吴奇. 基于带有噪声输入的稀疏高斯过程的人体姿态估计. *自动化学报*, 2019, **45**(4): 693–705)
- 10 Qiao Shao-Jie, Han Nan, Ding Zhi-Ming, Jin Che-Qing, Sun Wei-Wei, Shu Hong-Ping. A multiple-motion-pattern trajectory prediction model for uncertain moving objects. *Acta Automatica Sinica*, 2018, **44**(4): 608–618 (乔少杰, 韩楠, 丁治明, 金澈清, 孙未未, 舒红平. 多模式移动对象不确定性轨迹预测模型. *自动化学报*, 2018, **44**(4): 608–618)
- 11 Wang Chuan-Yun, Qin Shi-Yin. Background modeling of infrared image in dynamic scene with Gaussian mixture model in compressed sensing domain. *Acta Automatica Sinica*, 2018, **44**(7): 1212–1226 (王传云, 秦世引. 动态场景红外图像的压缩感知域高斯混合背景建模. *自动化学报*, 2018, **44**(7): 1212–1226)
- 12 Da B S, Ong Y S, Gupta A, Feng L, Liu H T. Fast transfer Gaussian process regression with large-scale sources. *Knowledge-Based Systems*, 2019, **165**: 208–218
- 13 Liu X, Ma Z M. Discriminant analysis with local Gaussian similarity preserving for feature extraction. *Neural Processing Letters*, 2018, **47**(1): 39–55
- 14 Binois M, Gramacy R B, Ludkovski M. Practical heteroscedastic Gaussian process modeling for large simulation experiments. *Journal of Computational and Graphical Statistics*, 2018, **27**(4): 808–821
- 15 Ren Zhi-Gang, Liang Yong-Sheng, Zhang Ai-Min, Pang Bei. A Gaussian estimation of distribution algorithm using general second-order mixed moment. *Acta Automatica Sinica*, 2018, **44**(4): 635–645 (任志刚, 梁永胜, 张爱民, 庞蓓. 基于一般二阶混合矩的高斯分布估计算法. *自动化学报*, 2018, **44**(4): 635–645)
- 16 Sarkar D, Osborne M A. Prediction of tidal currents using Bayesian machine learning. *Ocean Engineering*, 2018, **158**: 221–231
- 17 Nguyen-Tuong D, Seeger M, Peters J. Model learning with local Gaussian process regression. *Advanced Robotics*, 2009, **27**(13): 1003–1012
- 18 Berg M D, Cheong O, Kreveld M V, Overmars M. *Computational Geometry: Algorithms and Applications*. Berlin: Springer-Verlag, 2008. 99–105
- 19 Rasmussen C E. *Gaussian Processes in Machine Learning*. Berlin: Springer-Verlag, 2003. 7–31
- 20 Nocedal J, Wright S. *Numerical Optimization*. Berlin: Springer-Verlag, 2006. 101–133
- 21 Bentley J L. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 1975, **18**(9): 509–517
- 22 Freidman J, Bentley J L, Finkel R A. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software*, 1977, **3**(3): 209–226
- 23 Otair M. Approximate k-nearest neighbour based spatial clustering using k-d tree. *International Journal of Database Management Systems*, 2013, **5**(1): 1–12
- 24 Park C, Huang J H Z. Efficient computation of Gaussian process regression for large spatial data sets by patching local Gaussian processes. *Journal of Machine Learning Research*, 2016, **17**(1): 6071–6099
- 25 Fang G, Wang X M, Wang K, Lee K H, Ho J D L, Fu H C, Fu D K C, Kwok K W. Vision-based online learning kinematic control for soft robots using local Gaussian process regression. *IEEE Robotics and Automation Letters*, 2019, **4**(2): 1194–1201
- 26 Alcalá-Fdez J, Fernández A, Luengo J. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 2011, **17**(2–3): 255–287
- 27 Beckers T, Umlauf J and Hirche S. Stable model-based control with Gaussian process regression for robot manipulators. *IFAC-PapersOnLine*, 2017, **50**(1): 3877–3884



肖文鑫 北京大学计算机学院博士研究生. 主要研究方向为软件工程和机器学习.

E-mail: wenxin.xiao@stu.pku.edu.cn
(XIAO Wen-Xin Ph.D. candidate at the School of Computer Science, Peking University. His research interest covers software engineering and machine learning.)



张文文 同济大学电子与信息工程学院博士研究生. 主要研究方向为传感器检测技术与测量系统. 本文通信作者.

E-mail: zhangwenwen_1203@163.com
(ZHANG Wen-Wen Ph.D. candidate at the College of Electronic and Information Engineering, Tongji University. His research interest covers sensor detection technology and measurement system. Corresponding author of this paper.)