

doi: 10.3969/j.issn.0490-6756.2019.06.002

含随机参数的偏微分方程的自适应高斯过程求解器

陈晨^{1,2,3}, 廖奇峰³, 王皓⁴

(1. 中国科学院上海微系统与信息技术研究所, 上海 200050; 2. 中国科学院大学, 北京 100049;
3. 上海科技大学信息科学与技术学院, 上海 201210; 4. 四川大学数学学院, 成都 610064)

摘要: 对于数值求解含随机参数的偏微分方程的问题, 本文基于以高斯过程为核心的求解器提出了一种自适应挑选训练数据的求解模型. 该模型从极少的初始训练数据集出发训练高斯过程求解器, 将参数池中预测方差指示变量最大的参数及其对应的偏微分方程的高精度解加入训练数据集中, 然后重复上述过程, 直到所训练出来的高斯过程求解器在测试数据集上达到所要求的精度. 此外, 本文还将该自适应模型在带有二维随机参数的扩散方程上进行测试, 结果表明所提出的自适应选点策略有效, 模型的预测准确度随着训练数据的增加而迅速提高, 最终只需要 40 个训练数据即可在测试数据集上达到要求的精度.

关键词: 含参偏微分方程; 高斯过程; 自适应; 动态选点

中图分类号: O242.1 **文献标识码:** A **文章编号:** 0490-6756(2019)06-0997-07

An adaptive Gaussian process emulator for partial differential equations with stochastic parameters

CHEN Chen^{1,2,3}, LIAO Qi-Feng³, WANG Hao⁴

(1. Shanghai Institute of Microsystem & Information Technology, Chinese Academy of Sciences, Shanghai 200050, China;
2. University of Chinese Academy of Sciences, Beijing 100049, China;
3. School of Information Science & Technology, Shanghai Tech University, Shanghai 201210, China;
4. School of Mathematics, Sichuan University, Chengdu 610064, China)

Abstract: This paper aims at the numerical solution of partial differential equations (PDEs) with stochastic parameters. We propose an emulator which adaptively chooses training data based on Gaussian process. This model begins with very limited number of training data and trains a Gaussian process emulator, then adds the parameter with the highest prediction variance indicator from the parameter pool along with its corresponding high-fidelity PDE output into the training data set. This process continues until the model achieves a desired accuracy on the test data set. A two-dimensional parametric diffusion equation is used to test this adaptive model. Numerical results demonstrate that the proposed adaptive data selection is effective. The accuracy of our model increases rapidly as the training data grow, and only 40 training data allow us to obtain a desired accuracy.

Keywords: Parametric partial differential equation; Gaussian process; Adaptive; Active point selection (2010 MSC 35C99; 65C20)

收稿日期: 2019-03-15

基金项目: 国家自然科学基金(11601329)

作者简介: 陈晨(1994-), 男, 主要从事不确定性量化方面的研究. E-mail: chenchen3@shanghaitech.edu.cn

通讯作者: 王皓. E-mail: wangh@scu.edu.cn

1 引言

偏微分方程在大气模拟、流体仿真以及石油勘测中有广泛应用。由于背景问题的复杂性,人们要求解的偏微分方程常带有随机参数。使用传统方法(有限差分方法^[1],有限元方法^[2]等)虽然可以得到高精度数值解,但这些方法通常在模型复杂度和计算复杂度上不占优势。因此,发展高效率高精度的替代模型一直是这方面研究的热点。

在过去的十几年里,对带随机参数的偏微分方程的替代模型以及模型的化简方法的研究有着巨大的进步。几类常用的方法如谱方法(spectral methods)^[3-4],高斯过程模型(Gaussian process)^[5-6]和人工神经网络(artificial neural networks)^[7-8]等都在不同的目标任务上有着良好的表现。但是,在实验设计方面,如在数据驱动模型中,如何选择训练数据目前的工作并不多。这个问题非常重要,尤其在求解高精度解极为耗时耗力的情况下。因此,高效地选择最适合模型的最优训练数据集不仅能够大大降低准备训练数据的时间、节省人力,还能够提高模型的运行速度且不降低模型的准确度,在现实应用场景中具有很强的现实意义。

受到简化基模型(reduced basis methods)^[9-10]以及动态学习模型^[11]的启发,本文基于已有的高斯过程求解器^[6]发展了一个动态选择训练数据的自适应模型,并通过实验来验证该自适应模型的有效性,说明了实验设计(有针对性的选择训练数据)的重要性。

2 问题描述

令 (Ω, Σ, P) 为一个完备概率空间,其中 Ω 为样本空间, $\Sigma \in 2^\Omega$ 为 σ 代数, $P: \Sigma \rightarrow [0, 1]$ 为概率测度。同时,定义 $D \in \mathbf{R}^s, s=1, 2, 3$ 为有界连通物理空间,其边界为 ∂D 。我们考虑问题:确定从 $D \times \Omega$ 到 \mathbf{R} 的映射 $u(x, \xi)$,使得

$$\begin{cases} L(x, \xi; u(x, \xi)) = f(x), \forall (x, \xi) \in D \times \Omega, \\ B(x, \xi; u(x, \xi)) = g(x), \forall (x, \xi) \in \partial D \times \Omega \end{cases} \quad (1)$$

成立,其中 $L(\cdot), B(\cdot)$ 分别为带有随机参数的微分算子和边界算子, $f(\cdot)$ 和 $g(\cdot)$ 分别为源函数和边界函数。对随机参数 ξ 的任一取值,若使用高精度数值解法来计算(1)式的解,可以得到其离散数值解

$$y = [u(x^{(1)}, \xi), \dots, u(x^{(N_h)}, \xi)]^T \in \mathbf{R}^{N_h} \quad (2)$$

其中 $u(x^{(i)}, \xi), i=1, \dots, N_h$ 代表 $u(x, \xi)$ 在某一空

间网格上的取值, N_h 代表空间自由度(网格点个数)。假设(1)式的离散数值解构成空间 $M \in \mathbf{R}^{N_h}$,则问题可以被改写成:生成一个偏微分方程求解器 P 来模拟(2)式中从 Ω 到 M 的映射,使得对随机参数 ξ 有

$$P(\xi) = \hat{y} \approx [\hat{u}(x^{(1)}, \xi), \dots, \hat{u}(x^{(N_h)}, \xi)]^T \quad (3)$$

本文的目的是基于已有的高斯过程求解器,寻求一个动态选择训练数据的方法,使得我们可以用少量的训练数据获得一个不牺牲精度的高斯过程求解器。

3 高斯过程求解器

这一部分将介绍基于核主成分分析^[12]的高斯过程求解器。假设已知一组训练数据 $\Theta: \{(\xi^{(i)}, y^{(i)})\}, i=1, \dots, N$ 。该求解器主要包含三个步骤:

- 1) 对输出空间进行非线性降维(核主成分分析),继而得到这批输出数据的降维表达;
- 2) 利用高斯过程回归模型对从输入数据到输出数据的降维表达的映射进行模拟,进而对于一个新的输入数据 ξ 可以预测与之对应的输出 y 的降维表达;
- 3) 利用逆向映射方法将预测的 y 的降维表达映射回输出数据空间,得到对 y 的预测 $P(\xi) = \hat{y}$ 。

该算法整体框架如图 1 所示。接下来我们对这三个步骤做必要的细节介绍。

3.1 核主成分分析

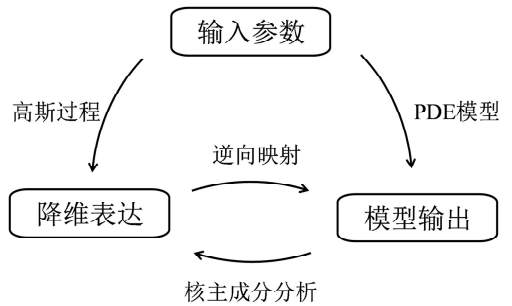


图 1 高斯过程求解器算法框架示意图
Fig. 1 Framework of the Gaussian process emulator

核主成分分析是传统主成分分析的一种非线性形式,因而基于传统主成分分析的结论可以被推广到核主成分分析。

给定训练数据 $\Theta: \{(\xi^{(i)}, y^{(i)})\}, i=1, \dots, N$,假定输出数据被投影 $\phi: \mathbf{R}^{N_h} \rightarrow F$ 映射到内积空间 F ,并被零均值化成为 $\{\tilde{\phi}(y^{(i)}), i=1, \dots, N\}$ 。接下来求解特征问题

$$\bar{C}v = \lambda v \quad (4)$$

其中 $\bar{C} = \frac{1}{N} \sum_{i=1}^N \tilde{\phi}(y^{(i)}) \tilde{\phi}(y^{(i)})^T$. (4) 式的任何非零

解 v ($\lambda \neq 0$) 都可由 $\tilde{\phi}(y^{(1)}), \dots, \tilde{\phi}(y^{(N)})$ 线性张成^[12]. 这意味着

$$v = \sum_{i=1}^N \alpha_i \tilde{\phi}(y^{(i)}) \quad (5)$$

同时, 考虑另外一组方程:

$$(\tilde{\phi}(y^{(k)}) \cdot \bar{C}v) = \lambda(\tilde{\phi}(y^{(k)}) \cdot v), k=1, \dots, N \quad (6)$$

将(5)式代入(6)式可以推导出

$$\tilde{K}^2 \alpha = N\lambda \tilde{K} \alpha \quad (7)$$

其中 $\tilde{K}_{ij} = \tilde{\phi}(y^{(i)}) \tilde{\phi}(y^{(j)})^T$, $\alpha = [\alpha_1, \dots, \alpha_N]^T$. (7)

式的解可通过求解特征值问题 $\tilde{K} \alpha = N\lambda \alpha$ 来获得^[12]. 令 $\{\lambda_i\}$, $i=1, \dots, N$ 为解得的降序排列的所有特征值并假设 λ_p 为最后一个非零特征值,

$\{\alpha_i\}$ 为对应的特征向量. 通过调整 α_i 的大小: $\tilde{\alpha}_i = \alpha_i / \sqrt{\lambda_i}$, 可以归一化 \bar{C} 的特征向量, 也即使得

$$\|\tilde{v}_i\|_2 = \left\| \sum_{k=1}^N \tilde{\alpha}_k \tilde{\phi}(y(k)) \right\|_2 = 1, i=1, \dots, p \quad (8)$$

令 y 为任一输出空间的数据. 可以通过

$$z_i(y) = (\tilde{v}_i \cdot \tilde{\phi}(y)) =$$

$$\sum_{j=1}^N \tilde{\alpha}_j (\tilde{\phi}(y(j)) \cdot \tilde{\phi}(y)), i=1, \dots, p \quad (9)$$

来计算 $\tilde{\phi}(y)$ 在方向 \tilde{v}_i 上的投影, 进而将 $\tilde{\phi}(y)$ 表示为

$$\tilde{\phi}(y) = \sum_{i=1}^p z_i(y) \tilde{v}_i \quad (10)$$

通过(7)式和(9)式可知, 整个计算过程需要空间 \mathbf{F} 中内积操作是明确定义的, 这也提醒我们该空间本身可以被隐式定义. 这就是所谓的“核技巧”——通过一个只牵涉到空间 \mathbf{R}^N 内部计算的核函数来定义空间 \mathbf{F} 中的内积操作

$$(\phi(y(i)) \cdot \phi(y(j))) = k(y(i), y(j)) \quad (11)$$

\mathbf{F} 因此被称为核导出希尔伯特空间 (reproducing kernel Hilbert space). 本文采用最常用的高斯核函数

$$k(y(i), y(j)) = e^{-\|y(i)-y(j)\|^2/s^2} \quad (12)$$

其中 s 为给定常数. 确定核函数之后, 我们可以通过 $K_{ij} = k(y(i), y(j))$ 来定义核矩阵 \mathbf{K} , 紧接着可以得到 $\tilde{\mathbf{K}} = \mathbf{H}\mathbf{K}\mathbf{H}$, 其中 $\mathbf{H} = \mathbf{I} - (1/N)(\mathbf{1}\mathbf{1})^T$, \mathbf{I} 为单位阵, $\mathbf{1}$ 为全 1 向量. 继而(9)式可被改写为

$$z_i(y) = (\tilde{v}_i \cdot \tilde{\phi}(y)) = \tilde{\alpha}_i^T \mathbf{H} \left(\mathbf{k} - \frac{1}{N} \mathbf{K} \mathbf{1} \right) \quad (13)$$

其中 $\mathbf{k} = [\phi(y(1)) \cdot \phi(y), \dots, \phi(y(N)) \cdot \phi(y)]^T$.

为了实现降维, 我们考虑利用(10)式中的部分项

$$\tilde{\phi}_r(y) = \sum_{i=1}^r z_i(y) \tilde{v}_i \quad (14)$$

来近似 $\tilde{\phi}(y)$, 其中 $r < p \leq N$. 该式的截断误差可以

通过 $\|\tilde{\phi}_r(y) - \tilde{\phi}(y)\|^2 = \sum_{i=r+1}^p \lambda_i^2$ 来计算^[13]. 因此 r

可以通过 $\sum_{i=1}^r \lambda_i / \sum_{i=1}^p \lambda_i > \epsilon$ 来选择, 其中 ϵ 是提前

设置的阈值. 本文中, 我们称(14)式右边项中所有系数排成的向量 $z(y) = [z_1(y), \dots, z_r(y)]^T$ 为输出数据 y 的降维表达.

3.2 高斯过程回归模型

对于一个新的输入 ξ , 因为 y 是未知的, 我们无法显式地得到它对应的输出 y 的降维表达, 而且“核技巧”定义的空间 \mathbf{F} 是隐式的. 为了解决这个问题, 高斯过程回归模型被引入来建立从输入 ξ 直接到输出 y 的降维表达的映射.

首先, 定义复合映射 η :

$$\eta(\xi) = z(y) = z(P(\xi)) = [\eta_1(\xi), \dots, \eta_r(\xi)]^T \quad (15)$$

由于 $\eta_1(\xi), \dots, \eta_r(\xi)$ 相互独立 (对应相互正交的特征向量), 它们每一个都由一个单独的高斯过程来回归, 即 $\eta_i(\xi) \sim GP(E(\xi), Cov(\xi, \xi'))$, 其中 $E(\xi)$ 和 $Cov(\xi, \xi')$ 分别代表该高斯过程均值函数和协方差函数.

一个高斯过程可以被看作是随机变量的集合, 该集合中任意数量的随机变量都符合一个联合高斯分布^[14]. 在我们的设定中, 随机变量就是函数 $\eta_i(\xi)$ 在不同参数 ξ 下的值. 本文取零均值函数 $E(\xi) = 0$ 和带噪声的平方指数协方差函数

$$Cov(\xi, \xi') = \rho_1^2 e^{-(\xi - \xi')^T \text{diag}(l_1, \dots, l_M) - 1} (\xi - \xi')/2 + \rho_2^2 \delta(\xi, \xi') \quad (16)$$

其中超参数 l_1, \dots, l_M 和 ρ_1^2, ρ_2^2 分别为平方相关长度以及信号方差, $\delta(\xi, \xi')$ 为克罗内克 δ 函数.

对于每一个高斯过程 $\eta_i(\xi)$, $i=1, \dots, r$, 超参数 $\theta = [l_1, \dots, l_M, \rho_1^2, \rho_2^2]^T$ 是通过最小化负对数边缘似然函数 $L(\theta)$:

$$L(\theta) = \frac{1}{2} \mathbf{t}_i^T \mathbf{C}^{-1}(\theta) \mathbf{t}_i + \frac{N}{2} \log(2\pi) \quad (17)$$

得到的, 其中 $\mathbf{t}_i = [\eta_i(\xi(1)), \dots, \eta_i(\xi(N))]^T$, $\mathbf{C}(\theta)$ 是协方差矩阵, 由 $\mathbf{C}(\theta)_{jk} = Cov(\xi(j), \xi(k))$, $j, k=1, \dots, N$ 定义. 最小化(17)式通常是一个非凸问

题^[15],一般利用经典梯度方法来解决,如共轭梯度法^[16]和拟牛顿法^[17].当超参数被确定下来后,我们可以得到 $\eta_i(\xi)$ 的概率分布预测:

$$\eta_i(\xi) | \xi, t_i, \xi(1), \dots, \xi(N) \sim GP(\bar{\eta}_i(\xi), \text{Cov}(\eta_i(\xi))) \quad (18)$$

其中

$$\begin{aligned} \bar{\eta}_i(\xi) &= c_*^T C(\theta) - \mathbf{1}t_i, \\ \text{Cov}(\eta_i(\xi)) &= \text{Cov}(\xi, \xi) - c_*^T C(\theta) - \mathbf{1}c_* \\ c_* &= [\text{Cov}(\xi, \xi(1)), \dots, \text{Cov}(\xi, \xi(N))]^T \end{aligned} \quad (19)$$

我们用 $\eta_i(\xi)$, $i=1, \dots, r$ 的均值作为对其的预测值,最终得到 $\hat{z}(y) = \hat{\eta}(\xi) = [\bar{\eta}_1(\xi), \dots, \bar{\eta}_r(\xi)]^T$.

3.3 逆向映射方法

通过前两步,我们可以得到一个新的输入 ξ 对应的输出 y 的降维表达 $\hat{z}(y)$. 接下来,我们介绍从降维表达 $\hat{z}(y)$ 恢复输出 y 的逆向映射方法. 一个未知的输出 y 可以被距离其最近的若干输出数据的加权平均逼近^[6]:

$$\hat{y} = \sum_{i=1}^n \omega_i y(n_i) \quad (20)$$

其中 $y(n_i)$, $i=1, \dots, n$ 是 y 的 n 个近邻, ω_i 是对应的权值. 本文中我们定义权值为

$$\omega_i = \frac{e^{-d(y, y(n_i))}}{\sum_{j=1}^n e^{-d(y, y(j))}}, i=1, \dots, n \quad (21)$$

其中 $d(y, y(n_i))$ 代表 y 和 $y(n_i)$ 之间的欧氏距离. 这种形式的权值比较稳定,结果比较精确^[6].

下面说明如何近似 $d(y, y(i))$, $i=1, \dots, N$. 首先定义 $\Phi = [\phi(y^{(1)}), \dots, \phi(y^{(N)})]$, 其零均值化形式为 $\tilde{\Phi} = [\tilde{\phi}(y^{(1)}), \dots, \tilde{\phi}(y^{(N)})] = \Phi H$. 在核空间中, $\phi(y)$ 和 $\phi(y^{(i)})$ 的距离 d_i 可以通过

$$d_i = 2 - \phi(y)^T \phi(y^{(i)}) = 2 - 2k(y, y^{(i)}) \quad (22)$$

来计算,其中 $k(\cdot, \cdot)$ 为核主成分分析中的高斯核函数. 同时,我们将

$$\begin{aligned} \phi_r(y) &= \sum_{i=1}^r z_i(y) \tilde{v}_i + \bar{\phi} = \sum_{i=1}^r z_i(y) \Phi H \tilde{\alpha}_i + \\ &\frac{1}{N} \Phi \mathbf{1} = \Phi (H[\tilde{\alpha}_1, \dots, \tilde{\alpha}_r] + \frac{1}{N} \mathbf{1}) = \Phi \tau \end{aligned} \quad (23)$$

代入(22)式中近似代替 $\phi(y)$, 得到

$$d_i \approx \tau^T K \tau + 1 - 2 \tau^T k_i \quad (24)$$

其中 $K = \Phi^T \Phi$ 为核矩阵, k_i 为核矩阵 K 的第 i 列.

联立(22), (24)两式,我们可以得到

$$d(y, y^{(i)}) \approx -s^2 \ln(0.5 - 0.5 \tau^T K \tau + \tau^T k_i) \quad (25)$$

通过以上计算可以得到 $d(y, y^{(i)})$, $i=1, \dots, N$ 的逼近值,再通过(20)(21)两式即可利用逆向映射方

法得到输出 y 的预测值 \hat{y} .

整个高斯过程求解器的算法流程如算法 3.1 所示.

算法 3.1 高斯过程求解器

Begin

初始化 训练数据 $\Theta: \{(\xi^{(i)}, y^{(i)})\}, i=1, \dots, N$, 任意测试输入参数 ξ ;

目标 求解器 P , 其满足 $P(\xi) = \hat{y}$;

步骤 1 核主成分分析

1) 计算中心化核矩阵 $\tilde{K} = HKH$;

2) 求解关于 \tilde{K} 的特征值问题, 调整 α_i 的大小;

3) 设定阈值 ϵ , 计算训练输出数据的降维表达;

步骤 2 高斯过程回归

1) 针对降维表达的每一维, 训练高斯过程回归模型 $\eta_j(\cdot) = z_j(P(\cdot)) = GP(E(\cdot), \text{Cov}(\cdot, \cdot)), j=1, \dots, r$;

2) 使用 $\hat{z}(y) = \hat{\eta}(\xi)$ 作为 y 的降维表达的预测值;

步骤 3 逆向映射方法

1) 使用(25)式近似 y 和 $y^{(i)}, i=1, \dots, N$ 的距离;

2) 通过(20)(21)两式得到对 y 的预测值 \hat{y} .

End.

4 自适应高斯过程求解器

为了得到训练数据 $\Theta: \{(\xi^{(i)}, y^{(i)})\}, i=1, \dots, N$, 我们需要计算给定的偏微分方程在参数 $\{\xi^{(i)}, i=1, \dots, N$ 上的高精度解 $\{y^{(i)}, i=1, \dots, N$. 若训练数据的数量 N 很大, 则准备训练数据将耗费大量时间. 为了提高效率, 一种可行的方案是从数据池中挑选一个最优子集 Θ_r 作为训练数据, 该子集包含的数据数量较少, 并且使用其训练出来的模型不丧失准确度. 这种选择技巧被称为动态数据选择^[18-19].

基于第 2 部分介绍的高斯过程求解器, 我们提出一种动态选择训练数据的方法. 在该方法引导下的高斯过程求解器我们称之为自适应高斯过程求解器. 其整体流程如算法 4.1.

算法 4.1 自适应高斯过程求解器

Begin

初始化 参数池 $R: \{\xi_i\}, i=1, \dots, N$, 初始训练参数集 $R_r = \{\xi^{(i)}, \xi^{(i)} \in R, i=1, \dots, c$, 初始训练数据集 $\Theta_r = \{(\xi^{(i)}, y^{(i)})\}, \xi^{(i)} \in R_r, i=1, \dots, c$, 误

差阈值 ϵ' , 测试样本 $T: \{(\xi_j^*, y_j^*)\}, j=1, \dots, t$;

目标 用于训练算法 3.1 的最优子集 Θ_{tr} 并求解器 **P**

步骤 1 使用 Θ_{tr} 作为训练数据, T 作为测试数据执行算法 3.1, 得到求解器 **P**;

步骤 2 计算求解器 **P** 在测试集 T 上的平均相对误差 $\bar{e}(T)$;

步骤 3 若 $\bar{e}(T) < \epsilon'$, 说明 Θ_{tr} 已让求解器 **P** 达到一定精确度, 算法结束, 否则进行下一步;

步骤 4 更新参数池 $R = R - R_{tr}$;

步骤 5 使用 Θ_{tr} 作为训练数据, R 作为测试数据执行算法 3.1 的步骤 1 和 2;

步骤 6 计算误差指示变量 $\gamma(\xi), \xi \in R$,

步骤 7 选择 ξ_{new} 使得 $\gamma(\xi_{new})$ 在 $\{\gamma(\xi)\}, \xi \in R$ 中最大, 并将其加入集合 R_{tr} 中, $R_{tr} = R_{tr} \cup \{\xi_{new}\}$;

步骤 8 计算高精度数值解 $y(\xi_{new}) = u_h(\xi_{new})$;

步骤 9 更新 $\Theta_{tr}, \Theta_{tr} = \Theta_{tr} \cup \{(\xi_{new}, y(\xi_{new}))\}$ 并回到步骤 1;

End.

在步骤 2 中, 平均相对误差 $\bar{e}(T)$ 被定义为

$$\bar{e}(T) = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{\|\hat{y}_i^* - y_i^*\|^2}{\|y_i^*\|^2} \quad (26)$$

其中 $|T|$ 表示集合 T 包含元素个数. 在动态选择训练数据的过程中, 我们使用高斯过程回归模型中的预测值的标准差来定义误差指示变量 $\gamma(\xi)$:

$$\gamma(\xi) = \sum_{i=1}^r \frac{\lambda_i}{\sum_{j=1}^r \lambda_j} sd^2(\eta_i(\xi)) \quad (27)$$

其中 $sd(\cdot)$ 表示随机变量的标准差, $\lambda_i, i=1, \dots, r$ 是(4)式中协方差矩阵 \bar{C} 的前 r 个最大的特征值. 这种误差指示变量的设置既考虑了高斯回归模型的预测准确程度, 也考虑了输出数据的低维表达的各个维度的重要程度. 基于高斯过程, 我们可以很方便地进行基于标准差的动态选点. 同时, 高斯过程对于预测值的不确定性的把握以及问题本身所带有的不确定性提供了将模型嵌入贝叶斯框架的可能性^[11].

5 数值实验

本文考虑定义在物理空间 $D = (-1, 1) \times (-1, 1)$ 上定义的带有分段系数的扩散方程

$$\begin{aligned} -\nabla \cdot [a(x, \xi) \nabla u(x, \xi)] &= 1, \text{ in } D \times \Gamma, \\ u(x, \xi) &= 0, \text{ on } \partial D_D \times \Gamma, \\ \frac{\partial u(x, \xi)}{\partial n} &= 0, \text{ on } \partial D_N \times \Gamma \end{aligned} \quad (28)$$

其中 $x = [x_1, x_2]$ 为物理空间中的任一网格点, $\frac{\partial u(x, \xi)}{\partial n}$ 是向外法方向导数, $\partial D_D \cap \partial D_N = \emptyset, \partial D = \partial D_D \cup \partial D_N$. 我们采用狄利克雷边界条件, 即 $\partial D_N = \emptyset$. $a(x, \xi)$ 为渗透系数, 在物理空间上分两段取值, 即

$$a(x, \xi) = \begin{cases} \xi_1, & x \in (-1, 0) \times (-1, 1), \\ \xi_2, & x \in (0, 1) \times (-1, 1) \end{cases} \quad (29)$$

其中 $\xi_i, i=1, 2$ 在 $[0, 2, 1]$ 上均匀分布. 该方程的高精度解由双线性有限元方法^[20-21] 求解.

为了测试算法 4.1, 我们利用蒙特卡洛方法生成了 $N=500$ 组输入参数 ξ , 其中 400 组作为参数池 R , 对于剩下的 100 组参数我们计算它们对应的高精度数值解, 继而用它们共同组成测试样本 T . 初始训练参数集 R_{tr} 包含两组输入参数, 从 R 中随机挑选获得. 其它实验超参数设置: 算法 3.1 中 $\epsilon = 0.9$, 算法 3.2 中 $\epsilon' = 0.01$, (20) 式中的 $n = \min(\lceil 0.5 |\Theta_{tr}| \rceil, 10)$. 同时, 为了避免数值不稳定性带来的算法的提前终止, 我们将算法 3.2 的步骤 3 加强为: 若在连续五次循环中 $\bar{e}(T) < \epsilon'$ 都成立, 说明当前训练数据集 Θ_{tr} 已让求解器 **P** 稳定地达到一定精确度, 算法结束; 否则进行下一步. 算法 4.1 初始化完毕后, 我们运行算法 4.1, 其自动终止在 $|\Theta_{tr}| = 40$. 我们记录下算法 4.1 的每一次循环中在测试数据集上的平均相对误差 $\bar{e}(T)$, 其变化如图 2 所示. 由该图能够看出, 随着算法 4.1 选择的训练数据的增加, 求解器 **P** 在测试数据集上的平均相对误差迅速地下降. 这不仅说明我们提出的自适应动态选点策略

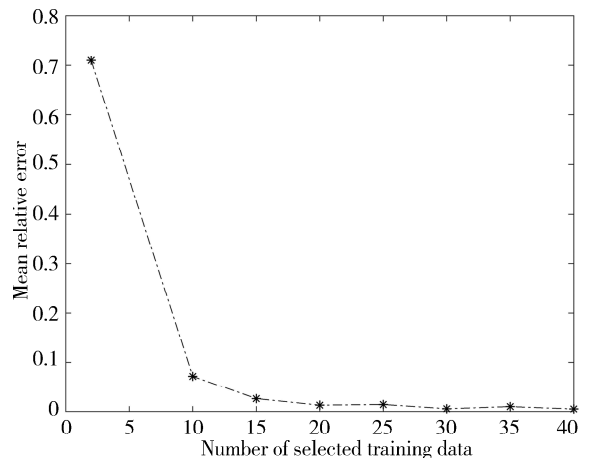


图 2 平均相对误差随训练数据增加的变化
Fig. 2 Decay of the average relative error as $|\Theta_{tr}|$ increases

在降低模型预测误差上是有效的,也说明了针对这个问题我们并不需要大量训练数据,因而从小数据集出发迭代搜索得到最优子训练集 Θ_r 的想法是正确的. 针对该带有二维随机参数的扩散方程,我们只需要 40 个训练数据(由算法 4.1 进行选择)就能使求解器达到要求的精度.

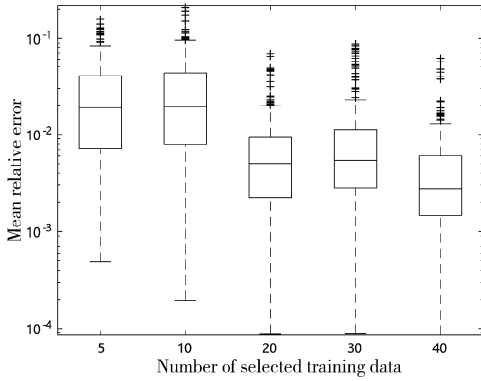


图 3 平均相对误差的箱型图

Fig. 3 Boxplots of the average relative error

图 3 是训练数据数量分别为 5, 10, 20, 30, 40 时平均相对误差的箱型图. 可以看出随着训练数据的增加,求解器的准确性在整体上升. 当算法 4.1 停止后,用所选择的 40 个训练数据训练出来的求解器在测试集上的相对误差主要集中在区间 $[10^{-3}, 5 \times 10^{-2}]$, 达到一个比较准确的预测水平. 同时我们也关心由算法 4.1 挑选出来的训练参数在参数

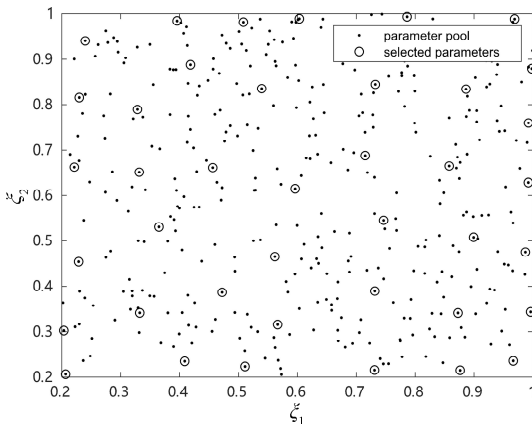


图 4 所选训练参数在参数空间中的分布

Fig. 4 Distribution of the selected training parameters

空间中的分布特点,如图 4 所示. 可以看出,被选择出来的参数在参数空间中整体呈均匀分布,其中有接近一半的参数点分布在靠近参数空间边界的地方,剩下的点较均匀地分布在参数空间的内部. 我

们认为,超参数 ϵ' 在一定程度上决定了所选择训练参数的分布密集程度, ϵ' 越小,算法 4.1 对模型预测精度要求越高,则所选择训练参数分布越密集,反之亦然. 图 5 展示了该模型在某一随机选取测试参数上的预测解与高精度解的对比.

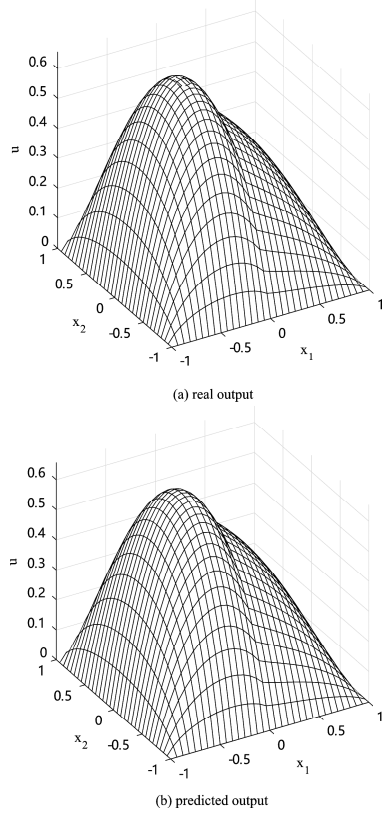


图 5 某测试参数上的预测解与高精度解的对比
Fig. 5 Comparison between the predicted output and the high-fidelity output of one test parameter

6 结论

数值实验表明,本文提出的自适应高斯过程求解器能够有效地动态添加训练数据,不断提高模型的预测准确度,最终得到数据总量不大的(相对)最优训练数据集,使在该数据集上训练的高斯过程求解器达到要求的精度,降低了模型复杂度.

该算法也存在提升的空间. 一个可供将来进一步研究的思路是,可否避免引入测试数据,仅仅利用参数池来指示算法的自动停止? 如果可行,一方面可以节省准备测试数据的时间,另一方面能够防止算法对测试数据的过拟合. 有针对性地设置预测误差指示变量 $\gamma(\xi)$, 使其不但可以定性比较参数池中不同输入参数上的预测误差大小,而且能够定量地指示当前求解器 \mathbf{P} 的精度,可能是一个可行的方案,也是未来我们工作的一个方向.

参考文献:

- [1] Morton K W, Mayers D F. Numerical solution of partial differential equations: an introduction [M]. Cambridge: Cambridge University Press, 2005.
- [2] Smith G D. Numerical solution of partial differential equations: finite difference methods [M]. Oxford: Oxford University Press, 1985.
- [3] Ma X, Zabaras N. An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations [J]. J Comput Phys, 2009, 228: 3084.
- [4] Powell C E, Elman H C. Block-diagonal preconditioning for spectral stochastic finite-element systems [J]. IMA J Numer Anal, 2009, 29: 350.
- [5] Raissi M, Perdikaris P, Karniadakis G E. Numerical Gaussian processes for time-dependent and nonlinear partial differential equations [J]. SIAM J Sci Comput, 2018, 40: A172.
- [6] Xing W W, Triantafyllidis V, Shah A A, *et al.* Manifold learning for the emulation of spatial fields from computational models [J]. J Comput Phys, 2016, 326: 666.
- [7] Raissi M, Perdikaris P, Karniadakis G E. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations [J]. J Comput Phys, 2019, 378: 686.
- [8] Sirignano J, Spiliopoulos K. DGM: a deep learning algorithm for solving partial differential equations [J]. J Comput Phys, 2018, 375: 1339.
- [9] Chen Y, Gottlieb S, Heryudon A, *et al.* A reduced radial basis function method for partial differential equations on irregular domains [J]. J Sci Comput, 2016, 66: 67.
- [10] Jiang J, Chen Y, Narayan A. Offline-enhanced reduced basis method through adaptive construction of the surrogate training set [J]. J Sci Comput, 2017, 73: 853.
- [11] Guo M, Hesthaven J S. Reduced order modeling for nonlinear structural analysis using Gaussian process regression [J]. Comput Method Appl M, 2018, 341: 807.
- [12] Schölkopf B, Alexander S, Müller K R. Nonlinear component analysis as a kernel eigenvalue problem [J]. Neural Comput, 1998, 10: 1299.
- [13] Wold S, Esbensen K, Geladi P. Principal component analysis [J]. Chemometr Intell Lab, 1987, 2: 37.
- [14] Williams C K, Rasmussen C E. Gaussian processes for machine learning [M]. Cambridge: MIT Press, 2006.
- [15] Snelson E L. Flexible and efficient Gaussian process models for machine learning [D]. United Kingdom: University College London, 2007.
- [16] Møller M F. A scaled conjugate gradient algorithm for fast supervised learning [J]. Neural Comput, 1993, 6: 525.
- [17] Shanno D F. Conditioning of quasi-Newton methods for function minimization [J]. Math Comput, 1970, 24: 647.
- [18] Beck J, Guillas S. Sequential design with mutual information for computer experiments (MICE): emulation of a tsunami model [J]. SIAM/ASA J Uncertain Quan, 2016, 4: 739.
- [19] Settles B. Active learning literature survey [R]. Madison, Wisconsin, United States: Department of Computer Sciences, University of Wisconsin-Madison, 2009.
- [20] Braess D. Finite elements: theory, fast solvers, and applications in solid mechanics [M]. Cambridge: Cambridge University Press, 2007.
- [21] Elman H C, Silvester D J, Wathen A J. Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics [M]. Oxford: Oxford University Press, 2014.

引用本文格式:

中文: 陈晨, 廖奇峰, 王皓. 含随机参数的偏微分方程的自适应高斯过程求解器[J]. 四川大学学报: 自然科学版, 2019, 56: 997.

英文: Chen C, Liao Q F, Wang H. An adaptive Gaussian process emulator for partial differential equations with stochastic parameters [J]. J Sichuan Univ: Nat Sci Ed, 2019, 56: 997.