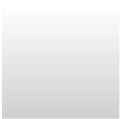
神经处理快报 (2021) 53:1147–1160 https://doi.org/10.1007/s11063-021-10434-9



**KTBoost：结合内核和树提升**

**法比奥·西格里斯特[一](http://orcid.org/0000-0002-3994-2244)**

接受日期：2021 年 1 月 23 日/在线发布日期：2021 年 2 月 11 日 © The Author(s) 2021

# 摘要

我们介绍了一种名为“KTBoost”的新型提升算法，它结合了**内核**提升和**树**提升。在每次增强迭代中，该算法将回归树或再生核希尔伯特空间 (RKHS) 回归函数添加到基学习器的集合中。直观上，这个想法是不连续的树和连续的 RKHS 回归函数相互补充，并且这种组合允许更好地学习具有不同规律性部分（例如不连续部分和平滑部分）的函数。我们根据经验表明，在对广泛的数据集进行比较时，KTBoost 在预测准确性方面明显优于树和内核提升。

**关键词**梯度和牛顿提升·再生核希尔伯特空间（RKHS）回归·集成学习·监督学习

# 简介

Boosting 算法 [ 8、15、17、18、28 ] 在应用数据科学和机器学习研究中享有很高的知名度，除此之外，由于它们在广泛的数据集上观察到的高预测准确性[ 11 ] 。Boosting 通过按顺序最小化风险函数来加性地组合基础学习器。尽管在 Freund 和 Schapire [ 15 ] 以及 Freund 和 Schapire [ 16 ] 的开创性论文中几乎没有对基学习器类型的限制，但关于组合不同类型基学习器的研究却很少。据我们所知，除了一个参考 [ 22]，现有的提升算法仅使用一种类型的函数作为基础学习者。迄今为止，回归树是基学习器最常见的选择，近年来人们付出了很多努力来开发可扩展到大数据的基于树的提升方法[ 11、26、35、36 ] 。

在本文中，我们通过组合回归树 [ 7 ] 和再现核希尔伯特空间 (RKHS) 回归函数 [ 4、39 ] 作为基础学习者，放宽了仅使用一种类型的基础学习者的假设。简而言之，RKHS 回归是非参数回归的一种形式，它显示了许多数据集的最先进的预测准确性，例如，它可以实现接近最优的测试误差 [1、2]和内核分类器平行于 Zhang 等人指出的深度网络的行为。[ 46 ]。由于现在越来越多的证据表明基础学习器不

乙

法比奥西格里斯特 fabio.sigrist@hslu.ch

1 卢塞恩应用科学与艺术大学，Suurstoffi 1, 6343 Rotkreuz, Switzerland

必然需要具有低复杂性 [ 44 ]、连续或平滑，RKHS 函数因此有可能补充不连续树作为基础学习者。

## 结果总结

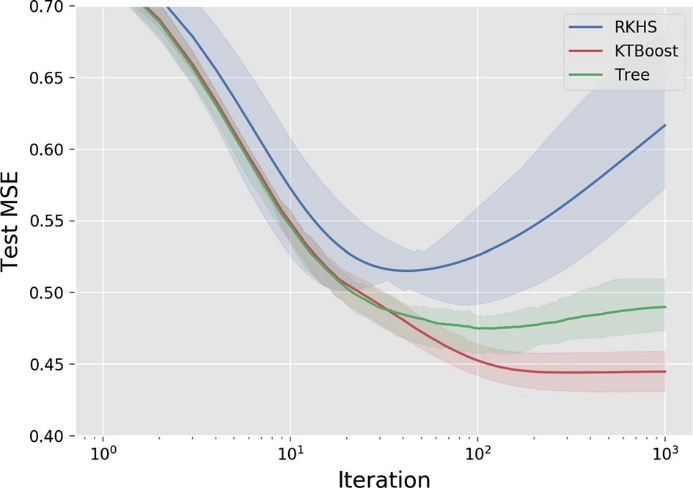
我们介绍了一种由“KTBoost”表示的新型提升算法，它结合了**内核**和**树**提升。在每次增强迭代中，KTBoost 算法添加回归树或惩罚 RKHS 回归函数，也称为核岭回归 [ 30]], 合奏. 这是通过首先使用功能牛顿法或功能梯度下降的一步学习树和 RKHS 函数，然后选择添加到集成中导致经验风险最低的基学习器来完成的。因此，KTBoost 算法在每次迭代中从两个根本不同的函数类中选择一个基础学习器。RKHS 中的函数是连续的，并且根据核函数的不同，它们也具有更高的规律性。另一方面，树是不连续的函数。

直觉上，这个想法是不同类型的基础学习者相互补充，并且这种组合可以更好地学习表现出不同程度规律性的部分的功能。我们在第 1 节的模拟研究中展示了这种效果。4.1 。为了简要说明树和 RKHS 函数作为基学习器的组合可以获得更高的预测精度，我们在图1中报告测试均方误差 (MSE) 与一个数据集（葡萄酒）的提升迭代次数。实线显示十个随机拆分为训练、验证和测试数据集的平均测试 MSE 与提升迭代次数的关系。逐点排除最大和最小 MSE 后获得置信带。所有方法的调整参数都是在验证数据集上选择的。见教派。4有关数据集和调整参数选择的更多详细信息。[[1]](#footnote-1)[1]该图说明了与树和内核提升相比，树和内核提升 (KTBoost) 的组合如何导致更低的测试 MSE。在我们在 Sect. 中的广泛实验中。4.2[[2]](#footnote-2) ，我们在大量数据集上展示了与仅树和仅内核提升相比，树和 RKHS 函数的组合导致较低的泛化错误。我们的方法在 Python 包KTBoost中实现，该包在 Python 包索引 (PyPI) 存储库中公开可用。[[3]](#footnote-3)[2][[4]](#footnote-4)

## 相关工作

结合来自多个模型的预测已成功应用于机器学习的许多领域，例如多样性诱导方法 [ 29 ] 或多视图学习；参见 Peng 等人。[ 34 ] 最近的一个增强应用程序示例。然而，boosting 结合基础学习器的方式不同于传统的集成，传统集成由几个在可能不同的数据集上训练的模型组成，因为例如，boosting 减少了方差和偏差。关于在提升框架中组合不同类型的基础学习器的研究很少，而且据我们所知，还没有研究调查提升不同类型的基础学习器时对预测准确性的影响。

mboost RpackageofHothornetal.[ 22 ] 允许组合不同的基础学习器，其中包括线性函数、一维和二维平滑样条、空间项、



**图 1**测试均方误差 (MSE) 与 KTBoost 的提升迭代次数与一个数据集（葡萄酒）的树和内核提升相比

回归树，以及用户定义的树。这种方法与我们的方法不同，因为 mboost使用组件方式的方法，其中每个基础学习器通常只依赖于几个特征，并且在每次增强更新中，添加了最小化对经验风险负梯度的最小二乘近似的项到乐团。相比之下，在我们的方法中，树和内核机器默认依赖于所有特征，基学习器是使用牛顿法或梯度下降法学习的，我们选择添加到集成中直接导致经验风险最低的基学习器。

机器学习方法应该能够学习平滑和非平滑函数的想法最近在机器学习的其他领域也受到了关注。例如，Imaizumi 和 Fukumizu [ 24 ] 以及 Hayakawa 和 Suzuki [ 20 ] 认为，深度神经网络比核方法具有更高预测准确性的原因之一是它们还能够学习非光滑函数。

# 预赛

## 提升

存在人口以及增强算法的样本版本。为了简洁起见，我们这里只考虑后者。假设我们有来自概率分布*P X* , *Y*的数据{( *x i* , *y i* ) ∈ R *p* × R, *i* = 1 ,..., *n* } 。boosting 的目标是找到函数*F* : R *p* → R用于预测给定*x的y*，其中*F*位于函数空间Ω S中，内积·,·由*F*给出，*F* = *E X*，期望是关于*P X* , *Y*的边际分布*P X*的。请注意，*y*可以是分类的、离散的、连续的或混合类型，具体取决于条件分布*P Y* | *X*关于勒贝格、计数测度或两者的混合是绝对连续的；参见，例如，Sigrist 和 Hirnschall [ 42] 以后者为例。根据数据和应用程序的目标，函数也可以是多变量的。为了符号简单起见，我们在下文中假设*F*是单变量的。多变量情况 *F* = ( *F k* ), *k* = 1 ,..., *d*的扩展很简单；参见，例如，Sigrist [ 41 ]。

boosting 的目标是找到经验风险函数*R* ( *F* )的最小值*F* ∗ (·) ：

*n*

 *F*∗( *L*(*yi*, *F*(*xi*)), (1)

*我*= 1

where *L* ( *Y* , *F* ) is an appropriately chosen loss function such as the squared error for regression or the logistic regression loss for binary classification, and Ω S = *span* ( S ) is the span of a set of base learners S = { *f j* : R *p* → R } 。Boosting 通过将更新*f m*迭代添加到当前估计*F m* − 1以顺序方式找到*F* \* ( · ):

|  |  |  |  |
| --- | --- | --- | --- |
| *F m* ( *x* ) = *F m* − 1 ( *x* ) + *f m* ( *x* ),  使得经验风险最小化 | *f m* ∈ S , | *米*= 1 ,...,*米*, | (2) |

*f m f* ). (3)

由于这通常无法明确地找到，因此使用近似最小化器。根据是否使用梯度或牛顿提升，更新*f m*要么作为负函数梯度的最小二乘近似值获得，要么通过应用对应于最小化二阶泰勒展开的函数牛顿法的一步来获得风险功能；见教派。3或 Sigrist [ 41 ] 了解更多信息。为了提高预测精度 [ 18 ]，通常会在更新方程中添加一个额外的收缩参数ν > 0：

*F m* ( *x* ) = *F m* − 1 ( *x* ) + ν *f m* ( *x* )。 (4)

## 再现核希尔伯特空间回归

假设*K* : R *d* × R *d* → R是正定核函数。然后存在一个具有内积的再生核希尔伯特空间 (RKHS) H ，使得 (i) 函数*K* (·, *x* )属于H对于所有*x* ∈ R *d*和 (ii) *f*对于所有*f* ∈ H 。 

假设我们有兴趣找到最小值

*n*

argmin ( *y i* − *f* ( *x i* )) 2 + λ *f* 2 , (5) *f* ∈ H *i* = 1 H

其中λ ≥ 0 是正则化参数。代表定理 [ 40 ] 然后指出存在一个唯一的最小化形式

*n*

*F*

*j* = 1

( 5 ) 可以写成

精氨酸*\_*

α∈R*n*

其中*y* = ( *y* 1 ,..., *y n* ) *T* , *K* ∈ R *n* × *n*且*K ij* = *K* ( *x i* , *x j* ) ，且α = (α 1 ,..., α *n* ) *T* . 取导数并将它们归零，我们发现显式解为

α = ( *K* + λ *I n* ) − 1 *y* ,

其中*I n*表示*n*维单位矩阵。

高斯过程回归和核回归之间有着密切的联系。( 5 )的解是以具有协方差函数*K*的零均值高斯过程的数据为条件的后验均值。此外，由于

*f* ( *x* ) = *k* ( *x* ) *T* ( *K* + λ *I n* ) − 1 *y* ,

在哪里

*k* ( *x* ) = ( *K* ( *x* 1 , *x* ),..., *K* ( *x n* , *x* )) *T* , (6)

核回归也可以解释为两层神经网络。

## 回归树

我们用T表示由回归树 [ 7 ]组成的空间。按照 Chen 和 Guestrin [ 11 ]中使用的符号，回归树由下式给出

*f T* ( *x* ) = w *s* ( *x* ) ,

其中*s* : R *p* → { 1 ,..., *J* } , w ∈ R *J* , *J* ∈ N表示树*f T* ( *x* )的终端节点数。*s*决定树的结构，即空间的划分，w表示叶子值。正如 Breiman 等人一样。[ 7 ]，我们假设*s*对空间的划分是一棵二叉树，其中划分中的每个单元格都是*R j* = ( *l* 1 , *u*1 ] × ··· × ( *l p* , *u p* ] ⊂ R *p*且−∞ ≤ *l m* < *u m* ≤ ∞且*s* ( *x* ) = *j*如果*x* ∈ *R j*。

# 结合内核和树提升

令*R* 2 ( *F m* − 1 + *f* )表示泛函，它与当前估计*F m* − 1下 ( 1 )中经验风险的二阶泰勒近似成正比：

*n*

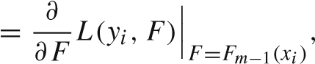
*Rh m* ,*如果(* x *i* ) 2 *,* ( 7)

2

*我*= 1

其中*g m* , *i*和*h m* , *i*是在函数*F m* − 1 ( *x* )和*I* { *x* = *x i* } ( *x* )处评估的经验风险的函数梯度和 Hessian ，其中*I* { *x* = *x i* } ( *x* ) = 1 如果*x* = *x i*否则为 0：

*米*,*我\_*

 2 (8)

∂ *h m* , *i*。

算法1中介绍的 KTBoost 算法的工作原理如下。在每次增强迭代中，候选树*f m T* ( *x* )和 RKHS 函数*f m K* ( *x* )被发现作为

牛顿法的函数版本的二阶泰勒近似。可以证明候选树*R* 2 ( *F m* − 1 + *f* ) 。这对应于应用一步*f mT* ( *x* )可以

被发现为加权最小二乘最小化；参见，例如，Chen 和 Guestrin [ 11 ] 或 Sigrist [ 41 ]。此外，可以找到候选惩罚RKHS回归函数*f m K* ( *x* ) ，如下面的命题1所示。然后，KTBoost 算法选择树或 RKHS 函数，以便将基础学习器添加到集成中

当量。( 4 ) 结果风险最低。请注意，对于 RKHS 增强部分，更新方程*F m* ( *x* ) = *F m* − 1 ( *x* ) + ν *f m* ( *x* )可以通过简单地更新系数α *m*来代替。

**Algorithm1:**

KTBoost

2:1:**初始化***m* = argmin *c* ∈R *d R* ( *c* ) 。

3: 计算梯度*g m* , *i*和 Hessian *h m* , *i*如 ( 8 )中所定义

4：找到候选回归树*f m T* ( *x* )和RKHS函数*f m K* ( *x* ) *f mT* ( *x* ) = argmin *f* ∈ T *R* 2 ( *F m* − 1 + *f* ) *f mK* ( *x* ) = argmin *f* 其中近似风险*R* 2 ( *F m* − 1 + *f* )定义为 (7)

5:**如果***R*  *F m***那么**

|  |  |
| --- | --- |
| 6: 7: | *f m* ( *x* ) = *f m T* ( *x* )  **别的** |
| 8: | *f m* ( *x* ) = *f m K* ( *x* ) |

9：**如果结束**

10： 更新 *F m* ( *x* ) = *F m* − 1 ( *x* ) + ν *f m* ( *x* )

11：**结束**

*如果损失函数在其第二个参数中不可微分或者二阶导数在X*的非空集上为零或常数，则可以选择使用梯度提升。KTBoost 的梯度提升版本作为算法1的一个特例，通过设置*h m* , *i* = 1 获得。梯度提升的优点是它在计算上比牛顿提升更便宜，因为与 ( 9 ) 相比，内核矩阵不依赖于迭代次数*m*；见教派。3.1了解更多详情。

**命题 1***正则化牛顿提升更新步骤中的核脊回归解 f m K* ( *x* )由 f *m K* ( *x* ) = *k* ( *x* ) *T* α *m给出*，*其中 k* ( *x* )*定义在*( 6 )*和*

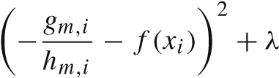
α *m* = *D m* ( *D m KD m* + λ *I n* ) − 1 *D m y m* , (9)

*其中 D m* = *diag , y m* = (− *g m* , 1 / *h m* , 1 ,...,− *g m* , *n* / *h m* , *n* ) *T*，*I n是维度 n 的单位矩阵。*

***证明***我们有

*n*

argmin λ *f* 2 *f* ∈ H *i* = 1 H

*n*

, *i f* 2 *f* ∈ H *i* = 1 H

*我是米*- *D米K*

如果我们对α取导数，使它们等于零，然后求解α ，我们发现

 *KD m* 2 *y m y m*

= *D m* ( *D m KD m* + λ *I n* ) − 1 *D m y m*。

## 降低大数据的计算成本

关于回归树，在生长树时找到​​分裂是计算要求很高的部分。关于如何有效地处理大数据，文献中有几种方法；参见，例如，Chen 和 Guestrin [ 11 ] 或 Ke 等人。[ 26 ]。查找内核回归更新的计算量大的部分是内核矩阵的因式分解，该矩阵随时间缩放为*O* ( *n* 3 ) 。有几种方法可以在大数据情况下提高计算效率。这方面的例子包括基于 Nyström 方法 [ 43 ] 的低秩近似及其扩展，例如分而治之核岭回归 [ 47], 48 ]、迭代优化方法 [ 6 、27 、38 、45 ] 的早期停止、随机梯度下降 [ 10 、12 ]、随机特征近似 [ 37 ] 和紧凑支持的核函数 [ 5 、19 ] 导致可以有效分解的稀疏核矩阵*K。*

请注意，如果使用梯度下降法而不是牛顿法，则可以通过观察有效地找到 RKHS 函数*f m K* ( *x* ) ，与 ( 9 )相比，核矩阵*K* +λ *I n*

不依赖于迭代次数*m*，即它的倒数或它的 Cholesky 因子只需要计算一次。此外，可以并行学习这两个学习器。

在我们的实证分析中，我们使用 Nyström 方法处理大型数据集。Nyström 方法通过首先选择一组*l*个所谓的 Nyström 样本*x* 1 ∗ ,..., *x l* ∗来近似内核*K* (·,·) 。通常这些是通过从数据中均匀抽样获得的。将与这些点对应的核矩阵表示为*K* ∗ ，然后 Nyström 方法将核*K* (·,·)近似为

*)*  ,

其中*k l* ( *x* ) *T*和*K j k l*。在

*特别地，全核矩阵K* 的降秩 Nyström 近似由下式给出

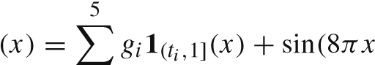
*K* ≈ *K n* ∗, *l K l* ∗, *l* − 1 *K n* ∗, *lT* ,

其中, 1 ≤ *j* ≤ *n* , 1 ≤ *k* ≤ *l*。

# 实验结果

## 模拟研究

我们首先进行了一个小型的模拟研究来说明不连续树和连续核机器的结合确实可以更好地学习具有不连续部分和光滑部分的函数。我们考虑随机函数*F* : [ 0 , 1 ] → R在[ 0 , 0中有五次随机跳跃。5 ] ：

 *女* ),

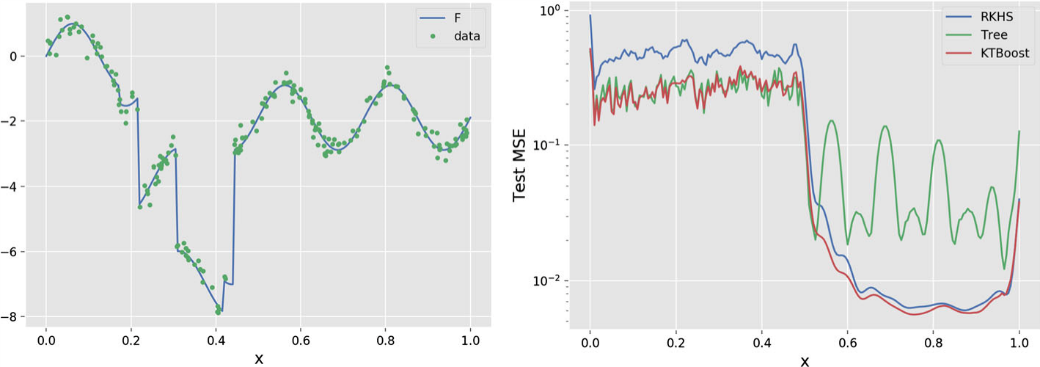
*我*= 1 (10)

二 二

*ti ∼* Unif ( 0 , 0 . 5 ) , *g i* ∼ Unif ( 0 , 5 )

和数据根据

*yi* = *F*(*xi*) + *N*(0,0.252), *xi* iid∼ Unif(0,1), *i* = 1,...,1000.



**图 2**在[ 0 , 0 ]中具有五次随机跳跃的随机函数示例。图 5 ]和相应的观测数据（左图）和树和内核提升的逐点均方误差 (MSE) 以及组合的 KTBoost 算法（右图）

在左侧的图2中，显示了此类函数和相应数据的示例。我们模拟 1000 次这样的随机函数以及大小为*n* = 1000 的训练、验证和测试数据。对于每次模拟运行，学习都是在训练数据上完成的。在验证数据上选择增强迭代次数，最大增强迭代次数为*M* = 1000。我们使用ν = 0的学习率。1 因为这是一个合理的默认值 [ 8 ] 并且树的深度= 1 因为没有交互。此外，对于 RKHS 岭回归，我们使用高斯核

*,* (11 )

其中ρ = 0 。1 和λ = 1。在右侧的图2中，我们显示了树和内核提升的逐点测试均方误差 (MSE) 以及组合的 KTBoost 算法。我们观察到 tree boosting 在不连续点所在的区域表现优于 kernel boosting，相反，kernel boosting 在平滑部分优于 tree boosting。该图还清楚地表明，KTBoost 优于树和内核提升，因为它在带有跳跃的区间上实现了树提升的 MSE，在平滑部分实现了内核提升的 MSE。

为了说明的目的，我们考虑了一个一维的例子。然而，在实践中，不连续性或强非线性以及平滑部分很可能发生在特征空间更高维度的交互级别。

## 真实世界数据

在下文中，我们使用以下 Delve、Keel、Kaggle 和 UCI 数据集将 KTBoost 算法与树和内核提升进行比较：鲍鱼、副翼、bank8FM、电梯、能源、住房、自由、NavalT、帕金森、puma32h、sarcos、葡萄酒、成人、癌症、ijcnn、电离层、声纳、汽车、癫痫、玻璃和 satimage。有关样本数量和特征的详细信息，请参见表1. 我们考虑回归以及二元和多类分类数据集。此外，我们包括不同大小的数据集，以研究较小和较大数据集的性能，因为尽管最近关注机器中的超大数据集，但中小型数据集继续广泛用于应用数据科学学习研究。我们将平方损失用于回归，将逻辑回归损失用于二元分类，将交叉熵损失与 softmax 函数用于多类分类。

|  |  |  |  |
| --- | --- | --- | --- |
| 数据 | # 类 | 注意。样品 | 注意。特征 |
| 鲍鱼 | 回归 | 4177 | 10 |
| 副翼 | 回归 | 13,750 | 40 |
| 银行8FM | 回归 | 8192 | 8 |
| 电梯 | 回归 | 16,599 | 18 |
| 活力 | 回归 | 768 | 8 |
| 住房 | 回归 | 506 | 13 |
| 自由 | 回归 | 50,999 | 117 |
| 海军 | 回归 | 11,934 | 16 |
| 帕金森病 | 回归 | 5875 | 16 |
| 彪马32h | 回归 | 8192 | 32 |
| 肉 | 回归 | 48,933 | 21 |
| 葡萄酒 | 回归 | 4898 | 11 |
| 成人 | 2 | 48,842 | 108 |
| 癌症 | 2 | 699 | 9 |
| ijcnn | 2 | 141,691 | 22 |
| 电离层 | 2 | 351 | 34 |
| 声呐 | 2 | 208 | 60 |
| 车 | 4 | 1728 | 21 |
| 癫痫的 | 5 | 11,500 | 178 |
| 玻璃 | 7 | 214 | 9 |
| 饱和图像 | 6 | 6438 | 36 |

**表 1**数据集汇总

对于回归数据集，我们使用梯度提升，对于分类数据集，我们使用带有牛顿更新的提升，因为这可以导致更准确的预测 [ 41 ]。对于某些分类数据集（成人、ijcnn、癫痫和 satimage），牛顿增强在当前 KTBoost 实现的标准单 CPU 计算机上在计算上是不可行的，尽管使用 Nyström 方法和合理数量的 Nyström 样本，例如1000，因为方程式中的加权核矩阵。( 9 ) 需要在每次迭代中分解。因此，我们也对这些数据集使用梯度提升。从技术上讲，这些案例可以使用牛顿方法或使用 Friedman 的混合梯度-牛顿提升版本来学习树 [ 18]，但这会导致不公平的比较，偏向于使用更好的优化方法学习的基础学习器。对于较大的数据集（liberty、sarcos、adult、ijcnn），我们使用第 1 节中描述的 Nyström 方法。3.1 。具体来说，我们使用 *l* = 1000 Nyström 样本，这些样本是从训练数据中均匀采样的。一般来说，Nyström 样本的数量越大，近似误差越低，但计算成本越高。Williams 和 Seeger [ 43 ] 报告了几个数据集的*l* ≈ 1000 的良好结果。所有计算均使用 Python 包KTBoost 完成在配备 2.9 GHz 四核处理器和 16 GB RAM 的标准笔记本电脑上。

将所有数据集随机分成三个大小相等的不重叠部分，得到训练集、验证集和测试集。学习是在训练数据上完成的，调整参数是在验证数据上选择的，模型比较是在 holdout 测试数据上完成的。使用训练数据对所有输入特征进行标准化，使其近似均值为零，方差为一。为了测量泛化误差并大致量化其中的可变性，我们将十种不同的数据随机拆分为训练、验证和测试集。我们注意到，当使用重采样方法时，标准统计测试（例如配对 t 测试）不能用于在数据集的基础上对不同算法进行成对比较，因为不同拆分中的训练和测试数据集由于重叠而相互依赖[3 , 13 , 14 ]。特别是，这会导致泛化误差的标准误差估计有偏差。

对于 RKHS 岭回归，我们再次使用高斯核；见等式。( 11 )。关于调整参数，我们从{ 1 , 2 ,..., 1000 }中选择提升迭代次数*M* ，从{ 1 , 10 − 1 , 10 − 2 , 10 − 3 }中选择学习率ν ，从{ 1 , 5中选择树的最大深度, 10 }和来自{ 1的核脊正则化参数λ, 10 } 。此外，内核范围参数ρ是使用*k*最近邻距离选择的，如下所述。我们首先计算训练数据中所有*k*最近邻的平均距离，其中*k*是从{ 5 , 50 , 500 , 5000 , *n* − 1 }中选择的调整参数，*n*是训练数据的大小。然后我们选择ρ使得核函数衰减到值 0 。01 在这个平均*k*- 最近邻距离。这是因为对于具有此类协方差函数的相应高斯过程，在该*k*最近邻距离处相关性已衰减至 1% 的水平。如果训练数据包含少于 5000（或 500）个样本，我们使用*n* - 1 作为*k 个*最近邻居的最大数量。此外，我们包括等于平均( *n* − 1 ) - 最近邻距离的ρ 。进行后一种选择是为了还包括一个范围，该范围导致内核在整个空间上缓慢衰减。对于使用 Nyström 方法的大数据集，我们计算平均*k*- 基于 Nyström 样本的最近邻距离。即，在这种情况下，最大的*k*等于*l* − 1。

结果如表2所示。对于回归数据集，我们显示了不同样本分割的平均测试均方误差 (MSE)，对于分类数据集，我们计算了平均测试错误率（=误分类率）。括号中的数字是不同样本分割的近似标准差。在最后一行，我们报告了每种方法在不同数据集上的平均排名。我们发现对于大部分数据集，KTBoost 比树和内核提升都实现了更高的预测准确性。具体来说，KTBoost 的平均排名为 1 。24，并且对于 21 个数据集中的 17 个，实现了比树和内核提升更高的预测精度。带有 Iman 和 Davenport 校正的 Friedman 检验 [ 25] 给出的*p*值为 7 。84 × 10 − 6这表明三种方法的差异非常显着。接下来，我们使用符号检验评估不同方法之间的成对准确性差异是否具有统计显着性。此外，我们应用 Holm–Bonferroni 校正 [ 21 ] 来解释我们进行多次测试的事实。*尽管符号测试具有低功效并且应用了保守的 Holm–Bonferroni 校正，但 KTBoost 比调整后的p*值低于 0 的树和内核提升要好得多。01. kernel 和 tree boosting 之间的差异并不显着，调整后和未调整的*p*值都在 0 以上。1（结果未制成表格）。

请注意，我们没有报告最佳调整参数，因为这对于数据集和样本拆分的所有组合都是不可行的，并且聚合值没有意义，因为不同的调整参数通常以非线性方式相互补偿（例如，迭代次数、学习率和树深度或核正则化λ )。此外，也很难根据不同的基学习器简明地总结集成的组成，因为在较早的提升阶段添加的基学习器比在后期添加的基学习器更重要 [9 ]]，并且基础学习器的属性也取决于所选择的调整参数。我们还注意到，还可以考虑额外的调整参数。对于树，这包括每叶、行和列子采样的最小样本数，

**表 2**使用测试均方误差（回归）和测试错误率（分类）比较 KTBoost 与树和内核提升

|  |  |  |  |
| --- | --- | --- | --- |
| 数据 | KTBoost | 树 | 核心 |
| 鲍鱼 | 4.65 (0.248) | 5.07 (0.261) | **4.64** (0.255) |
| 副翼 | **2.64e** - **08** (6.19e - 10) | 8.11e - 08 (2.39e - 09) | 2.64e - 08 (6.19e - 10) |
| 银行8FM | **0.000915** (4.02e - 05) | 0.000945 (2.47e - 05) | 0.000945 (5.83e - 05) |
| 电梯 | **4.83e** - **06** (2.9e - 07) | 5.66e - 06 (1.44e - 07) | 5.18e - 06 (3.89e - 07) |
| 活力 | **0.282** (0.0372) | 0.335 (0.093) | 1.3 (0.377) |
| 住房 | **12.7** (3.19) | 15.1 (3.23) | 13.6 (2.51) |
| 自由 | **14.5** (0.323) | 14.5 (0.314) | 15.2 (0.345) |
| 海军 | **6.51e** - **09** (1.15e - 09) | 1.15e - 06 (1.58e - 07) | 6.51e - 09 (1.15e - 09) |
| 帕金森病 | **73.3** (1.98) | 81.1 (2.44) | 73.3 (1.91) |
| 彪马32h | **6.5e** - **05** (2.27e - 06) | 6.51e - 05 (2.13e - 06) | 0.000695 (2.2e - 05) |
| 肉 | **7.99** (0.206) | 9.6 (0.207) | 17.8 (0.586) |
| 葡萄酒 | **0.444** (0.012) | 0.471 (0.0169) | 0.506 (0.0106) |
| 成人 | 0.128 (0.00295) | **0.128** (0.00313) | 0.163 (0.00512) |
| 癌症 | 0.0362 (0.00744) | 0.0415 (0.0153) | **0.0358** (0.0107) |
| ijcnn | **0.0122** (0.000685) | 0.0123 (0.000702) | 0.0387 (0.00516) |
| 电离层 | **0.0872** (0.017) | 0.103 (0.0226) | 0.107 (0.0239) |
| 声呐 | 0.194 (0.0394) | 0.223 (0.05) | **0.193** (0.0491) |
| 车 | **0.0399** (0.00505) | 0.0411 (0.00685) | 0.041 (0.00624) |
| 癫痫的 | **0.354** (0.00612) | 0.373 (0.00614) | 0.442 (0.0265) |
| 玻璃 | **0.308** (0.0711) | 0.315 (0.0589) | 0.344 (0.0581) |
| 饱和图像 | **0.089** (0.00452) | 0.112 (0.00504) | 0.0903 (0.00417) |
| 平均排名 | 1.24 | 2.48 | 2.29 |
| *p* val Friedman 检验 | 7.84e - 06 |  |  |
| 调整。*p* val 符号检验 |  | 6.29e - 05 | 0.00885 |

最小值以粗体显示。括号中是近似标准偏差。以下是不同数据集上方法的平均排名。*还报告了用于比较不同算法的*具有 Iman 和 Davenport 校正的 Friedman 检验的 p 值。最后一行显示了 Holm–Bonferroni 校正符号测试的*p*值，用于 KTBoost 算法与树和内核提升的成对比较

和休假值的惩罚，对于核回归，这包括核函数的平滑度，或者一般来说，核函数的类。也可以对两种类型的基础学习器使用不同的学习率。由于计算成本的限制，我们没有考虑调整参数的所有可能选择和组合。然而，树或内核提升中预测性能的潜在提高也可能会导致组合 KTBoost 算法的准确性提高。我们还注意到，在我们的实验设置中，与树和内核增强案例相比，KTBoost 算法的调整参数网格更大。不过，为了进行最公平的比较，这似乎是不可避免的。限制组合版本的一种类型的调优参数，但不限制单一基础学习者的情况似乎是别无选择的。在某种程度上缓解这种担忧的事实是，在上述模拟研究中，我们还发现在不使用交叉验证选择调整参数时表现出色，而不利的一面是，较大的调整参数网格也可能导致过度拟合。最后，我们指出，我们还考虑过比较未阻尼基学习器的风险

*R*

在算法1的第 5 行选择添加到集成中的基学习器时，我们获得了非常相似的结果（参见补充材料）。

# 结论

我们引入了一种新的提升算法，它结合了树和 RKHS 函数作为基础学习器。直观上，这个想法是不连续的树和连续的 RKHS 函数相互补充，因为树更适合学习函数的粗糙部分，而 RKHS 回归函数可以更好地学习函数的平滑部分。我们将 KTBoost 算法的预测精度与树和内核提升进行了比较，发现与树和内核提升相比，KTBoost 的预测精度要高得多。

未来的研究可以从几个方向进行。首先，除了树和核回归函数之外，研究其他基础学习器（如神经网络 [23、31 ] ）在多大程度上有用会很有趣。使用再生内核 Kre˘ın空间(RKKS) 学习器推广 KTBoost 算法[ 32、33] 而不是 RKHS 学习者也可以被调查。此外，学习率或风险界限等理论结果可能有助于进一步了解为什么树和内核机器的组合会导致预测准确性提高。最后，比较 KTBoost 算法在非常大的数据集上使用不同的策略来降低 RKHS 部分的计算复杂性会很有趣。在第 1 节中简要概述了关于如何做到这一点的几种潜在策略。3.1 。

**补充信息** 在线版本包含可在[https://doi 上获得的补充材料。组织/10.1007/s11063-021-10434-9 。](https://doi.org/10.1007/s11063-021-10434-9)

**致谢**感谢 Christoph Hirnschall 对本文提供的有用反馈。这项研究得到了瑞士创新机构-Innosuisse (25746.1 PFES-ES) 的部分支持。

Hochschule Luzern 提供的资金开放获取**资金**

**开放获取**本文根据知识共享署名 4.0 国际许可获得许可，该许可允许以任何媒体或格式使用、共享、改编、分发和复制，只要您对原作者和来源给予适当的信任，提供知识共享许可的链接，并说明是否进行了更改。本文中的图像或其他第三方材料包含在文章的知识共享许可中，除非在材料的信用额度中另有说明。如果材料未包含在文章的 Creative Commons 许可中，并且您的预期用途不被法律法规允许或超出允许的用途，您将需要直接从版权所有者那里获得许可。要查看此许可证的副本，请访问[http://creativecommons.org/licenses/by/4.0/.](http://creativecommons.org/licenses/by/4.0/)

# 参考

1. Belkin M, Hsu DJ, Mitra P (2018a) 过拟合还是完美拟合？插值的分类和回归规则的风险界限。在：Bengio S，Wallach H，Larochelle H，Grauman K，Cesa-Bianchi N，

Garnett R (eds) 神经信息处理系统的进展，第 31 卷。第 2306–2317 页

1. Belkin M, Ma S, Mandal S (2018b) To understand deep learning we need to understand kernel learning.In: Dy J, Krause A (eds) 第 35 届机器学习国际会议论文集，机器学习论文集第 80 卷学习研究。第 541–549 页
2. Bengio Y, Grandvalet Y (2004) No unbiased estimator of variance of k-fold cross-validation。J MachLearn Res 5:1089–1105
3. Berlinet A, Thomas-Agnan C (2011) Reproducing kernel Hilbert spaces in probability and statistics.Springer, Berlin
4. Bevilacqua M、Faouzi T、Furrer R、Porcu E 等人 (2019) Estimation and prediction using generalizedWendland covariance functions under fixed domain asymptotics。安统计 47(2):828–856
5. Blanchard G, Krämer N (2010) 内核共轭梯度回归的最优学习率。在：神经信息处理系统的进展中。第 226–234 页
6. Breiman L、Friedman J、Stone CJ、Olshen RA (1984) 分类和回归树。CRC出版社，博卡拉顿
7. Bühlmann P, Hothorn T (2007) 增强算法：正则化、预测和模型拟合。统计科学 22:477–505
8. Bühlmann P, Yu B (2003) 使用 l 2 损失提升：回归和分类。J Am Stat Ass98(462):324–339
9. Cesa-BianchiN,ConconiA,GentileC(2004)Onthegeneralizationabilityofon-linelearningalgorithms.IEEE Trans Inf Theory 50(9):2050–2057
10. Chen T, Guestrin C (2016) Xgboost：一种可扩展的树增强系统。在：第 22 届 acmsigkdd 知识发现和数据挖掘国际会议论文集中。美国计算机学会，第 785–794 页
11. Dai B, Xie B, He N, Liang Y, Raj A, Balcan M-FF, Song L (2014) 通过双随机梯度的可扩展核方法。在：神经信息处理系统的进展。第 3041–3049 页
12. Demšar J (2006) 多个数据集上分类器的统计比较。J Mach Learn Res 7:1–30
13. Dietterich TG (1998) 用于比较监督分类学习算法的近似统计检验。神经计算 10(7):1895–1923
14. Freund Y, Schapire RE (1996) 使用新的增强算法进行实验。见：ICML，第 96 卷。意大利巴里，第 148–156 页
15. Freund Y, Schapire RE (1997) 在线学习的决策理论推广和提升应用。J Comput Syst Sci 55(1):119–139
16. Friedman J、Hastie T、Tibshirani R 等人 (2000) Additive logistic regression: a statistical view of boosting（作者进行了讨论和反驳）。安统计 28(2):337–407
17. Friedman JH (2001) Greedy function approximation: a gradient boosting machine。安统计 29:1189–1232
18. Gneiting T (2002) 紧凑支持的相关函数。J Multivar 肛门杂志 83(2):493–508
19. Hayakawa S, Suzuki T (2020) 关于深度神经网络学习在稀疏参数空间上的极小极大最优性和优越性。神经网络 123:343–361
20. Holm S (1979) 一个简单的顺序拒绝多重测试程序。Scand J Stat 6:65–70
21. Hothorn T、Bühlmann P、Kneib T、Schmid M、Hofner B (2010) 基于模型的提升 2.0。J Mach LearnRes 11:2109–2113
22. Huang F, Ash J, Langford J, Schapire R (2018) Learning deep resnet blocks using boostingtheory sequentially using boostingtheory。ICML 80:2058–2067
23. Imaizumi M, Fukumizu K (2019) 深度神经网络有效地学习非光滑函数。在：第22届人工智能与统计国际会议。第 869–878 页
24. Iman RL, Davenport JM (1980) Fbietkan 统计临界区的近似值。CommunStat 理论方法 9(6):571–595
25. Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY (2017) Lightgbm：一种高效的梯度提升决策树。在：神经信息处理系统的进展。第 3149–3157 页
26. Ma S, Belkin M (2017) 潜入浅滩：大规模浅层学习的计算视角。在：神经信息处理系统的进展。第 3778–3787 页
27. Mason L、Baxter J、Bartlett PL、Frean MR (2000) 作为梯度下降的提升算法。在：神经信息处理系统的进展。第 512–518 页
28. Mendes-Moreira J、Soares C、Jorge AM、Sousa JFD (2012) 回归的集成方法：一项调查。ACM Comput Surv (CSUR) 45(1):10
29. Murphy KP (2012) 机器学习：概率视角。麻省理工学院出版社。书号 0262018020,9780262018029
30. Nitanda A, Suzuki T (2018) 基于残差网络感知的函数梯度提升。ICML80:3819–3828
31. Oglic D, Gaertner T (2018) Learning in reproducing kernel kre˘ı spaces。在：国际机器学习会议。第 3859–3867 页
32. Ong CS、Mary X、Canu S、Smola AJ (2004) 用非正核学习。在：第 21 届国际机器学习会议论文集。第 81 页
33. PengJ,AvedAJ,SeetharamanG,PalaniappanK(2018)Multiviewboostingwithinformationpropagationfor classification。关于神经网络和学习系统的 IEEE 汇刊 29(3):657–669
34. PonomarevaN,RadpourS,HendryG,HaykalS,ColthurstT,MitrichevP,GrushetskyA(2017)Tfboostedtrees：一种可扩展的基于张量流的梯度提升框架。在：关于数据库中机器学习和知识发现的欧洲联合会议。施普林格，第 423–427 页
35. Prokhorenkova L、Gusev G、Vorobev A、Dorogush AV、Gulin A (2018) Catboost：具有分类特征的无偏提升。载于：神经信息处理系统的进展第 31 卷。Curran Associates, Inc, pp 6638–6648
36. Rahimi A, Recht B (2008) 大规模内核机器的随机特征。在：神经信息处理系统的进展。第 1177–1184 页
37. Raskutti G、Wainwright MJ、Yu B (2014) 提前停止和非参数回归：最佳数据相关停止规则。J Mach Learn Res 15(1):335–366
38. Schölkopf B, Smola AJ (2001) Learning with kernels: support vector machines, regularization, optimization, and beyond。麻省理工学院出版社，剑桥
39. Schölkopf B、Herbrich R、Smola AJ (2001) 广义表示定理。在：计算学习理论国际会议。施普林格，第 416–426 页
40. Sigrist F (2021) 用于分类和回归的梯度和牛顿提升。专家系统应用程序 (inpress)
41. Sigrist F, Hirnschall C (2019) Grabit：用于默认预测的梯度树增强 tobit 模型。J BankFinance 102:177–192
42. Williams CK, Seeger M (2001) 使用 Nyström 方法加速内核机器。在：神经信息处理系统的进展。第 682–688 页
43. Wyner AJ、Olson M、Bleich J、Mease D (2017) 解释 adaboost 和随机森林作为插值分类器的成功。J Mach Learn Res 18(48):1–33
44. Yao Y, Rosasco L, Caponnetto A (2007) 关于梯度下降学习中的早期停止。Constr Approx26(2):289–315
45. Zhang C、Bengio S、Hardt M、Recht B、Vinyals O (2017) 理解深度学习需要重新思考泛化。在：关于学习表示的国际会议
46. Zhang Y, Duchi J, Wainwright M (2013) 分而治之核岭回归。在：学习理论会议。第 592–617 页
47. Zhang Y, Duchi J, Wainwright M (2015) 分而治之核岭回归：一种具有极小极大最优率的分布式算法。J Mach Learn Res 16(1):3299–3340

**出版商说明**施普林格·自然 (Springer Nature) 对已出版地图和机构隶属关系中的管辖权声明保持中立。

1. 为了更好地比较，收缩参数ν，参见等式。(4在本例中设置为固定值 (ν = 0.在 Sect. 的实验中。4.2，收缩参数也是使用交叉验证来选择的。 [↑](#footnote-ref-1)
2. [↑](#footnote-ref-2)
3. 有关详细信息，请参阅<https://github.com/fabsig/KTBoost> [↑](#footnote-ref-3)
4. [↑](#footnote-ref-4)