

多核贝叶斯优化的模型决策树算法*

高虹雷¹, 门昌骞¹, 王文剑^{1,2}

(1. 山西大学 计算机与信息技术学院, 山西 太原 030006;
2. 山西大学 计算智能与中文信息处理教育部重点实验室, 山西 太原 030006)

摘要:构造模型决策树时超参数较多,参数组合复杂,利用网格搜索等调参方法将会消耗大量的时间,影响模型性能的提升。提出了一种多核贝叶斯优化的模型决策树算法,该算法为应对不同分类数据特性,采用三种高斯过程建模寻优,利用贝叶斯优化技术,选出最优的参数组合。实验结果表明,所提算法在参数寻优上要优于传统的模型决策树寻优方法,并且能够在迭代次数不多的情况下找到全局最优参数值,在一定程度上提升了算法的分类性能,节省了大量的调参时间。

关键词:模型决策树;贝叶斯优化;高斯过程

中图分类号:TP391 文献标志码:A 文章编号:1001-2486(2022)03-067-10

Algorithm for model decision tree with multi-kernel Bayesian optimization

GAO Honglei¹, MEN Changqian¹, WANG Wenjian^{1,2}

(1. School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China;
2. Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Shanxi University, Taiyuan 030006, China)

Abstract: In the construction of the model decision tree, there are many parameters and the parameter combination is complex. The use of grid search and other parameter tuning methods will consume a lot of time, which will affect the improvement of the model performance. A model decision tree with multi-kernel bayesian optimization was proposed. In order to deal with the characteristics of different classified data, three Gaussian processes were used for modeling optimization. The Bayesian optimization technique was used to select the best parameter combination. The experimental results show that the proposed algorithm is better than the traditional model decision tree method in parameter optimization, and can find the global optimal parameter value in the case of a few iterations. To a certain extent, it improves the classification performance of the algorithm and saves a lot of parameter adjustment time.

Keywords: model decision tree; Bayesian optimization; Gaussian process

在机器学习中,模型要想达到一个不错的预测结果始终离不开超参数的合理设置。在当前大数据背景下,模型的超参数越来越多,比如神经网络^[1]、支持向量机^[2]、随机森林^[3](random forest, RF)以及极端梯度提升树^[4](extreme gradient boosting, XGBoost)、轻量级梯度提升机^[5](light gradient boosting machine, LightGBM)等广泛应用于工程领域的模型,这些模型在现实应用中由此带来的问题也层出不穷,比如组合过多,空间复杂,调参费时费力,还可能导致过拟合等问题。针对以上现状,如何高效地进行超参数优化是机器学习领域中至关重要的一个方面。一些算法提出

了有效的参数优化方法,例如 Falkner 等^[6]结合贝叶斯优化和赌博机方法的优点提出了一种快速收敛的参数确定方法。Narayan 等^[7]提出了一个用于超参数优化的开源分布式框架,用户可以灵活定制参数优化方法。此外,Liu 等^[8]提出一种通用的超参数优化框架,它使用户可以在分布式设置中使用所有可用的计算资源进行模型训练。而贝叶斯优化^[9](Bayesian optimization, BO)作为当前超参数优化算法中的主流方法,在自适应机器学习算法^[10]、强化学习^[11]、自然语言处理^[12]、推荐系统^[13]、游戏设计^[14]、传感器应用^[15]、环境监测^[16]、医学^[17]等应用广泛,相较经验调参方式

* 收稿日期:2020-06-08

基金项目:国家自然科学基金资助项目(62076154, U1805263);中央引导地方科技发展资金资助项目(YDZX20201400001224);山西省国际科技合作重点研发计划资助项目(201903D421050);山西省自然科学基金资助项目(201901D111030)

作者简介:高虹雷(1994—),男,山西忻州人,硕士研究生,E-mail:87133875@qq.com;
王文剑(通信作者),女,教授,博士,博士生导师,E-mail:wjwang@sxu.edu.cn

以及网格搜索^[18]、随机搜索^[19]等调参算法,贝叶斯优化算法大量降低了训练模型的时间,并且能够自动选择出最优参数,取得了很好的成绩。近年来,对贝叶斯优化算法的研究主要集中在概率代理模型以及效用函数的改进方面^[20]。例如, Srinivas 等^[21]利用置信边界策略重新构造效用函数,接着 Desautels 等^[22]提出批处理置信边界策略使实验可以分批进行。

传统上,多数调参过程凭借经验以及对算法的深刻理解,通过手动选参调参使得模型拥有一个良好的性能,但这种反复实验的方法无法保证能有效地找到最合适的超参数组合,且会耗费大量时间。近年来,超参数优化^[23]技术一直被业界积极探索,网格搜索即带有步长的穷举搜索,也是参数选择比较常用的算法之一,但在参数过多的情况下,该算法将会耗费大量的时间,在计算代价上呈指数级上升。而随机搜索则对超参数分布随机采样,且 Bergstra 和 Bengio 通过实验证明随机搜索算法^[19]比网格搜索更加高效快捷。虽然网格搜索和随机搜索在算法思想上比较简单,在给定的参数范围中测试,以此来确定最优值,理论上网格搜索一定可以找到全局最优值,但在面临参数较多、范围较广的实际情况下,计算代价特别庞大,且两者每次寻找新参数的过程都是独立于之前寻参过程的,即忽略先验分布,随机搜索虽然比网格搜索效率高一些,但同样也无法保证可以得到最优结果。

本文提出一种基于高斯过程的多核贝叶斯优化的模型决策树(model decision tree based on multi-kernel Bayesian optimization, MBOMDT)算法,针对模型决策树算法在构造模型决策树时超参数较多,参数组合复杂,利用网格搜索寻找最优参数费时费力,且不一定能够找到合适的参数值,设计了三种不同类型的核函数作为模型协方差函数组合,以此来应对不同分类数据的特性,实验表明所提算法在参数寻优上要优于传统的模型决策树(model decision tree, MDT)寻优方法,在一定程度上提升了算法的分类性能,节省了大量的调参时间。

1 MBOMDT 算法

1.1 MDT 算法

文献[24]提出的 MDT 算法的主要思想为在分类树叶子规模到达给定阈值时停止树的构建,并在生成的不完全决策树的可辨识结点(非叶结点且包含不同类别的样本)上搭载分类器,然后

通过分类器继续进行分类,图 1 为模型决策树的构建示意图。

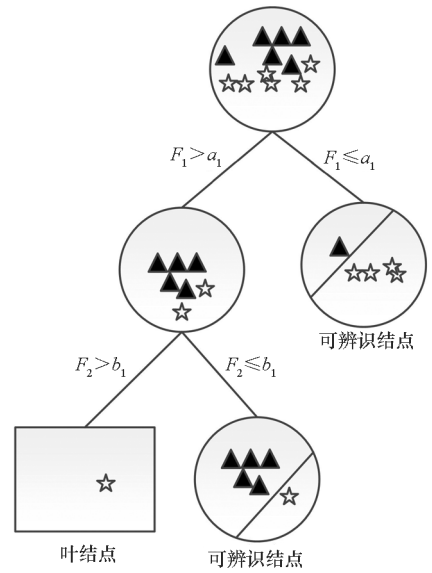


图 1 模型决策树构建示意

Fig. 1 Construction of model decision tree

1.2 多核贝叶斯优化算法

贝叶斯优化核心算法包含两个部分,即概率代理模型及效用函数(utility function)。在 MDT 算法中,叶子规模以及模型超参数的设置严重影响模型的性能。为了减少参数搜索的复杂度以及提升参数选择的稳定性,本文提出了一种 MBOMDT 算法。

如给定模型决策树 MDT,将叶子规模(参数)视为变量,用 x 表示。将 MDT 的分类错误率当作目标函数 $f(x)$,则贝叶斯优化的目标是找到恰当的 x 使得 $f(x)$ 最小。与网格搜索或随机搜索不同,贝叶斯优化在选择下一个观测点时考虑了已有的观测点。具体来说,给定 x 的已有观测点集合 $X = \{x_1, x_2, \dots, x_n\}$ 以及目标函数值集合 $Y = \{y_1, y_2, \dots, y_n\}$,贝叶斯优化利用集合 Y, X 以及概率代理模型拟合真实目标函数 $f(x)$,之后根据效用函数得到下一个观测点,并将观测点及其目标函数值分别放入集合 X 与 Y 中,更新概率代理模型。重复上述过程,直到满足终止条件,此时得到最优参数值。

本文使用一种常用的非参数型概率代理模型:高斯过程^[25]来拟合目标函数 $f(x)$ 。它由均值函数 $M(x)$ 及协方差函数 Σ 构成,可以表示为:

$$f(x) \sim GP(M(x), \Sigma) \quad (1)$$

假设集合 X, Y 在忽视噪声的情况下,即 $y = f(x)$, Y 服从联合高斯分布 $N(M(X), \Sigma)$, 并且协

方差函数可由核矩阵 $\mathbf{K}(X, X')$ 替代,公式如下:

$$\mathbf{K}(X, X') = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix} \quad (2)$$

其中,函数 $k(\cdot)$ 可以是任意核函数。为了表达的简洁,预处理样本使得均值 $M(x) = 0$ 。对于新的样本点 x_* 及其目标函数值 y_* ,高斯过程认为 $\begin{bmatrix} Y \\ y_* \end{bmatrix}$ 仍然服从联合高斯分布,具体如下所示:

$$\begin{bmatrix} Y \\ y_* \end{bmatrix} \sim N\left(0, \begin{bmatrix} \mathbf{k}(X, X) & \mathbf{k}(X, x_*) \\ \mathbf{k}(x_*, X) & \mathbf{k}(x_*, x_*) \end{bmatrix}\right) \quad (3)$$

$\mathbf{k}(X, x_*)$ 表示 x_* 与 X 之间的 3×1 阶核矩阵。通过以下公式可以形式化地表达上述过程:

$$p(y_* | X, Y, x_*) = N(M_*, \Sigma_*) \quad (4)$$

其中, $\Sigma_* = \mathbf{k}(x_*, x_*) - \mathbf{k}(x_*, X)\mathbf{k}(X, X)^{-1}\mathbf{k}(X, x_*)$, $M_* = \mathbf{k}(x_*, X)\mathbf{k}(X, X)^{-1}Y$ 。最终,通过上述过程可以对目标函数进行拟合。

效用函数则起到了确定下一估计点的作用,常用的效用函数有:提升概率最大评估点^[26](probability of improvement, POI)、最大化期望增量^[27](expected improvement, EI)、最大化高斯过程置信区间^[28](GP upper confidence bound, GP-UCB)。其中,GP-UCB 表达式为:

$$GP-UCB(X) = \mu(x) + \sqrt{\beta}\sigma(x) \quad (5)$$

参数 β 用来权衡均值及协方差,本文实验中选择的效用函数即 GP-UCB。

由式(1)可知,构建函数分布的能力以及理想的预测结果很大程度上依赖于协方差函数,即核函数的选取。此外,对于多信息源、多模态数据,如果可以结合多个单核函数,可以提升模型辨识能力,即多核学习算法^[29]。本文运用多核学习算法的思想,通过使用不同的核函数来提升算法性能。实验所使用的径向基函数(radial basis function, RBF)核函数分别表示为:

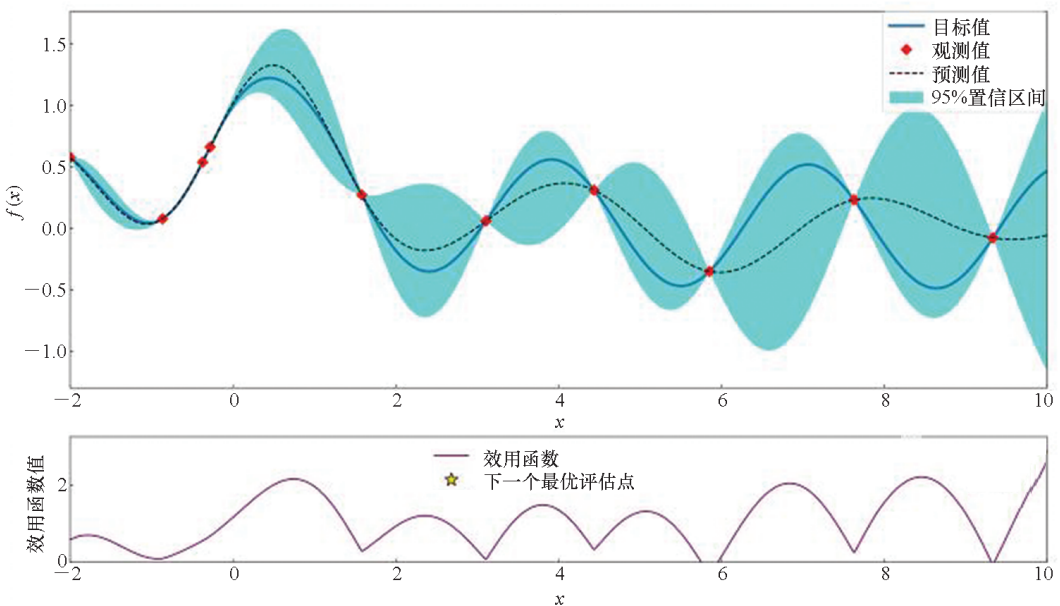
$$k(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{2l^2}\right) \quad (6)$$

$$k(x_i, x_j) = \left(1 + \frac{\sqrt{5}}{l}d(x_i, x_j) + \frac{5}{3l}d(x_i, x_j)^2\right) \cdot \exp\left(-\frac{\sqrt{5}}{l}d(x_i, x_j)\right) \quad (7)$$

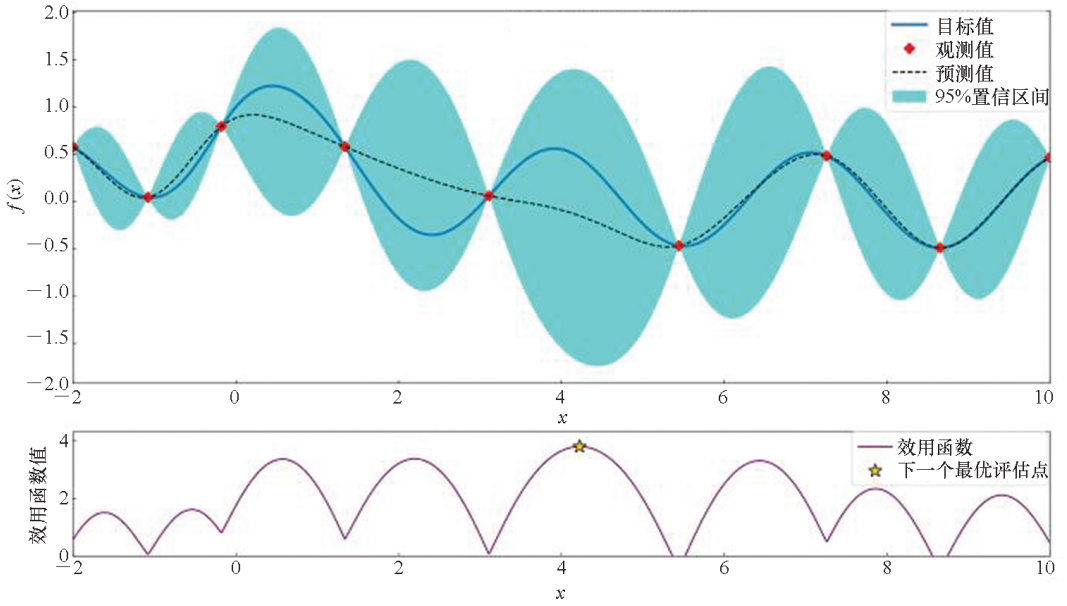
$$k(x_i, x_j) = \sigma_0^2 + x_i \cdot x_j \quad (8)$$

为更好地理解贝叶斯优化过程,本文给出一个例子进行说明,假设目标函数为 $f(x) = \sin x \cos x + 1/(x^2 + 1)$,图2分别为使用 RBF 核、Matern 核以及 DotProduct 核的贝叶斯优化过程示意图,每个实验迭代 10 次。

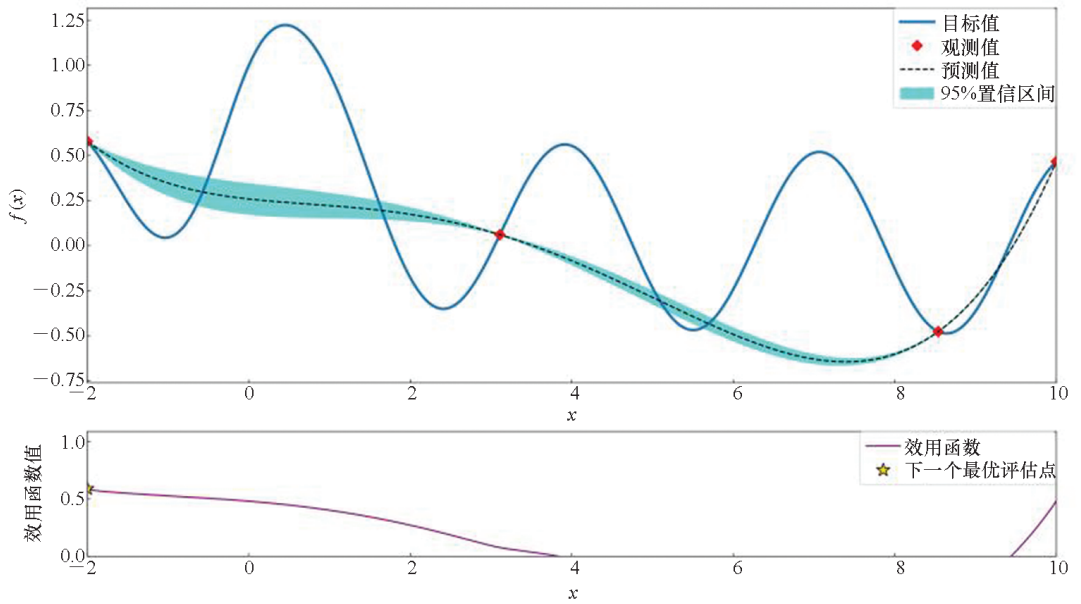
由图2可以看出,在后验分布中方差越小的点代表越接近真实分布,接下来通过最大化效用



(a) RBF 核
(a) RBF kernel



(b) Matern 核
(b) Matern kernel



(c) DotProduct 核
(c) DotProduct kernel

图 2 贝叶斯优化模型

Fig. 2 Bayesian optimization model

函数,将下一评估点选出,即图 2 中星点所在的位置,通过多次迭代过程,找到目标函数的最大值也即待调参数全局最优值。通过图 2 中不同核函数对同一目标函数进行优化可以得出,在寻优的过程中,使用不同核函数拟合出的后验分布具有不同的分布特性,当使用核函数组合来进行这一操作时,可以更大程度地覆盖没有被考虑的情况,提升算法整体的性能。多核贝叶斯优化算法如算法 1 所示。

2 实验结果及分析

2.1 实验数据集及实验设置

实验所用 13 个分类数据集均来自 UCI 数据集,如表 1 所示。计算机硬件及环境配置为: Intel(R) Core(TM) i7-7700 3.60 GHz 处理器,内存为 16.0 GB, Windows 10 操作系统, Python 3.7 版本。

算法1 多核贝叶斯优化算法

Alg.1 Multi-kernel Bayesian optimization algorithm

输入:目标函数 f ,超参数范围 X ,效用函数 α ,随机初始点个数 n ,迭代次数 N

输出: x_* , y_*

1. 随机初始化超参数集合: $D = \{x_1, x_2, \dots, x_n\}$,并得到相应的目标函数值集合 $Y = \{y_1, y_2, \dots, y_n\}$
2. 利用高斯过程 GP_R, GP_M, GP_D 对观测数据进行拟合,得到后验分布
3. 令效用函数 α 最大化,根据后验分布计算下一采样点
 $x_R = \arg \max \alpha(x), x_M = \arg \max \alpha(x), x_D = \arg \max \alpha(x)$
4. 计算不同核函数下采样点的目标值 $y_R = f(x_R), y_M = f(x_M), y_D = f(x_D)$
5. 计算并评估下一个采样点及其目标函数值:
 $x_{n+1} = \arg \min \{f(x_R), f(x_M), f(x_D)\}$
 $y_{n+1} = \min \{y_R, y_M, y_D\}$
6. 更新观测值集合 $X_{n+1} = X \cup \{x_{n+1}\}$,更新 GP_R, GP_M, GP_D
7. 重复执行步骤2~6,直到满足迭代次数 N

表1 实验数据集

Tab.1 Datasets used in experiments

UCI 数据集	训练 样本数	测试 样本数	属性 个数
Australian	390	300	14
Madelon	600	2 000	500
German	1 500	3 500	20
Breast_cancer	2 000	770	9
Spambase	2 500	1 500	57
Diabetes	3 000	4 680	8
Splice	5 000	10 875	60
Image	6 500	5 050	18
Banana	8 000	1 000	2
Credit Card Clie	9 000	20 000	23
Magic Gamma Telescope	9 020	10 000	10
Cod_rna	10 000	20 000	8
Flare_solar	20 000	33 300	9

MBOMDT 算法选用了软件包 LIBLINEAR 模型和标准支持向量机 (support vector machine, SVM) 模型,创建出三种新的模型,其中使用了 LIBLINEAR 模型的记为 MBOMDT_LIB,使用线性核的 SVM 模型记为 MBOMDT_SVM(linear),使用高斯核的 SVM 模型记为 MBOMDT_SVM(rbf)。

实验中的超参数分别为可辨识结点样本个数

阈值 V 、LIBLINEAR 中的类型参数 s 、SVM 中惩罚参数 c 、高斯核参数 γ 。实验内容主要为通过改进的贝叶斯优化算法对模型决策树超参数进行调参,以期得到直观明确的最优参数值,其中因为 LIBLINEAR 模型的参数 s 为类型参数,故在涉及使用 LIBLINEAR 模型的算法中类型参数 s 统一设置为 2,不再另做讨论。实验中所用到的 6 种算法的相关参数如表 2 所示,MDT 相关算法实验参数设置来自文献[24]。

表2 各算法实验参数一览表

Tab.2 Parameter of different algorithms

算法	s	c	γ	V
MDT_LIB	√			√
MDT_SVM(linear)		√		√
MDT_SVM(rbf)		√	√	√
MBOMDT_LIB	√			√
MBOMDT_SVM(linear)		√		√
MBOMDT_SVM(rbf)		√	√	√

注:“√”表示有此参数。

文献[24]中提出的三种模型决策树算法 MDT_LIB、MDT_SVM(linear) 和 MDT_SVM(rbf) 的超参数较多、调参过程费时费力,且三种算法已与 SVM、KNN、LR(logistic regression)、LIBLINEAR 等几种常用的分类方法进行了比较,实验结果表明文献[24]中提出的三种模型决策树算法在分类错误率及运行时间上大多优于其他方法,因此,本文所提三种模型将与文献[24]中提出的三种模型决策树算法以及原始贝叶斯优化下的模型决策树 BOMDT_LIB 算法、BOMDT_SVM(linear) 算法、BOMDT_SVM(rbf) 算法进行比较。

2.2 算法参数的优化

MDT 算法通过实验探究了可辨识结点样本个数阈值 V 对实验结果的影响,即调整参数 V 的大小以期找到实验结果的最优值,并作为算法的最优参数值进行下一步的分类研究。其方案为令可辨识结点样本个数阈值 V 分别取样本数的 10%、30%、50%、70%、90% 并在不同数据集上进行实验,找到相对较优参数值,SVM 模型中的惩罚参数 c 、高斯核参数 γ 均为人工调参并结合经验设定的固定值。其中:MDT_LIB 算法除在 Credit Card Clie 和 Cod_rna 数据集上 V 取训练样本数的 30%,在 Spambase 数据集上 V 取训练样本数的 90% 外,其余算法中 V 均为训练样本数的 10%。MDT_SVM(linear) 算法除在 Spambase 和

Cod_rna 数据集上 V 取训练样本数的 50%，在 Credit Card Clie 数据集上 V 取训练样本数的 70% 外，其余算法中 V 均为训练样本数的 10%。MDT_SVM(rbf) 算法除在 Germen 数据集上 V 取训练样本数的 50%，在 Image、Magic Gamma Telescope 以及 Cod_rna 数据集上 V 取训练样本数的 70%，在 Credit Card Clie 数据集上 V 取训练样本数的 90% 以外，其余算法中 V 均为训练样本数的 10%。使用线性核 SVM 模型中的惩罚参数 c 均为 1，使用高斯核 SVM 模型算法中的惩罚参数 c 均为 1、高斯核参数 γ 均为 0.5。相应地，本文通过改进贝叶斯优化算法对模型决策树进行参数选择，各参数范围分别是： $V \in (0, 100)$ 、 $c \in (1, 50)$ 、 $\gamma \in (0.01, 1)$ ，算法首先选取 3 个初始点，总计迭代次数为 20 次。通过实验为每种模型优化后的参数如表 3 所示。

在参数寻优的过程中，通过实验可以看出：各算法基本迭代次数为 10 次以内就可将综合范围内的最优参数值找到。因此，利用改进的贝叶斯优化方法寻优可大幅减少原算法训练最优参数的时间，提高算法分类效率。当使用 LIBLINEAR 模型时，算法参数为 V 。不同算法下运行时间比较如表 4 所示。MDT 算法从 $\{1, 5, 10, \dots, 100\}$ 中寻找 V 的最优值，MBOMDT 算法从 $(1, 100)$ 中寻找 V 的最优值，从表 4 中可以看到 MBOMDT 算法选择参数的时间明显低于 MDT 算法。另外，当使用高斯核 SVM 模型时，MDT 算法分别从 $\{1, 5, 10, \dots, 100\}$ 、 $\{1, 10, 20, \dots, 50\}$ 、 $\{0.01, 0.1, 0.2, \dots, 1\}$ 中寻找 V, c, γ 的最优值，而 MBOMDT 算法从 $(1, 100)$ 、 $(1, 50)$ 、 $(0.01, 1)$ 中寻找参数值，从表 4 还可以看出 MBOMDT 的参数寻优时间远低于 MDT，其中“/”表示时间超于 24 h。

表 3 MBOMDT 算法下参数的最优值

Tab. 3 Optimal value of parameters in MBOMDT

数据集	MBOMDT_LIB		MBOMDT_SVM(linear)		MBOMDT_SVM(rbf)		
	$V/\%$	$V/\%$	c	$V/\%$	c	γ	
Australian	3.659	99.790	2.330	38.360	1.358	0.834 4	
Madelon	10.570	17.920	0.875	19.970	1.066	0.507 0	
Germen	2.622	48.320	48.410	55.670	8.306	0.244 9	
Breast_cancer	1.000	1.000	23.230	1.133	30.030	0.133 0	
Spambase	62.190	66.400	1.479	46.940	12.310	0.040 2	
Diabetis	1.000	1.000	6.510	47.440	27.520	0.436 5	
Splice	1.011	1.307	49.560	47.660	48.170	0.022 8	
Image	14.660	36.900	46.920	92.420	1.018	0.521 0	
Banana	1.278	3.612	40.310	3.900	1.032	0.503 0	
CCC	25.750	69.510	1.874	20.580	5.126	0.010 0	
MGT	7.974	16.140	49.660	95.550	16.310	0.117 9	
Cod_rna	33.670	34.250	19.830	79.780	45.600	0.078 41	
Flare_solar	6.929	3.851	42.600	52.190	8.117	0.809 8	

表 4 不同算法下运行时间比较结果

Tab. 4 Comparison results of running time under different algorithms

单位：s

数据集	MDT		MBOMDT	
	BLINEAR	SVM(rbf)	LIBLINEAR	SVM(rbf)
Australian	11.02	143.72	5.38	5.68
Madelon	7 658.29	/	3 129.99	2 998.29
Germen	46.73	628.50	22.04	25.24
Breast_cancer	9.74	174.20	4.16	7.02
Spambase	1 435.69	16 422.00	324.97	561.37
Diabetis	82.71	1 249.62	28.47	39.59
Splice	208.73	3 488.93	74.79	100.63
Image	1 406.46	16 902.13	535.86	512.88
Banana	730.82	9 573.00	287.96	315.35
CCC	14 061.47	/	4 815.73	4 443.88
MGT	6 213.19	/	2 286.14	2 552.06
Cod_rna	281.45	5 031.48	126.79	158.58
Flare_solar	136.86	15 862.20	51.84	572.92

2.3 本文算法与 MDT 算法的比较

本节将本文提出的三种算法 MBOMDT_LIB、MBOMDT_SVM(linear)以及 MBOMDT_SVM(rbf)分别与文献[24]中的 MDT_LIB、MDT_SVM(linear)和 MDT_SVM(rbf)三种算法进行分类性能的比较,实验结果如表5所示。MDT 三种算法在部分数据集上的实验结果来自文献[24],本文

所提算法中的最优超参数设定均依照表3所得结果设置。由表5可知,本文所提三种算法在得到最优参数值之后使得算法的分类性能与之前对应的 MDT 算法相比,均得到了提升,其中 MBOMDT_SVM(rbf)算法在所有算法中表现最好,表中标黑数据为该数据集下6种算法分类错误率最小值。

表5 不同算法下分类错误率比较结果

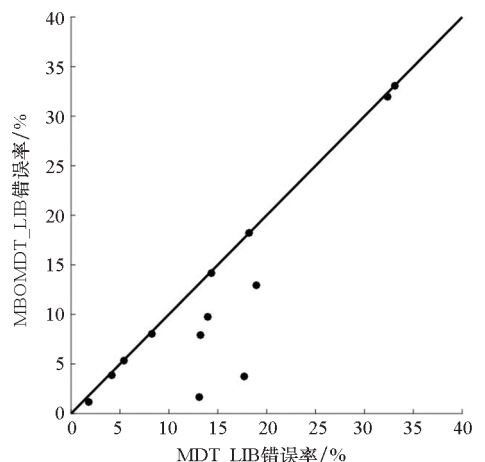
Tab.5 Classification error rates comparison of different algorithms

数据集	MDT			MBOMDT		
	LIBLINEAR	SVM(linear)	SVM(rbf)	LIBLINEAR	SVM(linear)	SVM(rbf)
Australian	19.00	20.33	16.33	13.000	12.330	13.660
Madelon	33.12	30.35	35.72	33.050	30.700	31.150
German	13.28	14.09	6.40	7.971	6.143	5.143
Breast_cancer	13.12	13.11	9.48	1.688	1.688	2.338
Spambase	8.21	8.20	15.40	8.067	7.933	9.733
Diabetis	17.72	17.70	4.26	3.733	3.833	0.366
Splice	1.77	1.63	1.55	1.177	1.039	0.855
Image	4.10	4.71	0.75	3.861	3.861	0.079
Banana	13.94	13.20	9.40	9.800	9.500	8.100
CCC	18.24	18.08	18.61	18.250	18.082	18.070
MGT	14.38	14.70	13.53	14.150	14.050	12.580
Cod_rna	5.40	5.34	6.23	5.300	5.190	4.915
Flare_solar	32.32	32.28	31.83	31.950	31.920	30.840

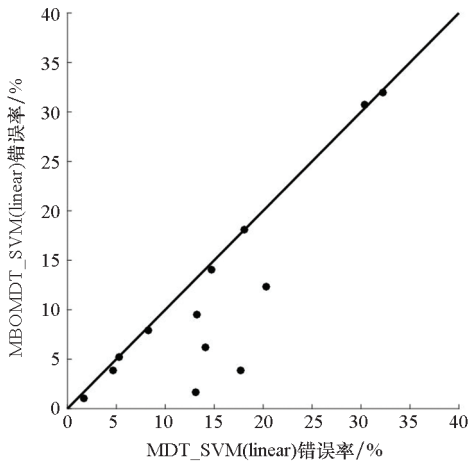
为直观地理解实验结果,图3给出了 MBOMDT 算法与 MDT 算法采用三种不同分类器时分类错误率的大小关系比较。其中,黑点表示所用的13个数据集,横坐标代表 MDT 算法的错误率,纵坐标代表 MBOMDT 算法的错误率。对角线上的黑点表示两种算法错误率一致或基本相同,在对角线上方的黑点表示在此数据集上 MBOMDT 算法错误率高于 MDT 算法错误率,对角线下方的黑点代表在此数据集上 MBOMDT 算法错误率低于 MDT 算法错误率。黑点距离对角线越远,表示在此数据集上两种算法的分类错误率相差越大。

由图3可知,当采用 LIBLINEAR 模型时,在5个数据集上 MBOMDT 的错误率明显低于 MDT 算法,且提升幅度较大,其余数据集上基本持平或稍好于原结果;当采用 SVM(linear)模型时,同样5个数据集上 MBOMDT 算法的错误率明显低于 MDT 算法,5个数据集上有小幅度提升,2个数据集基本持平,只有在1个数据集上略低于 MDT 算法;当采用 SVM(rbf)模型时,

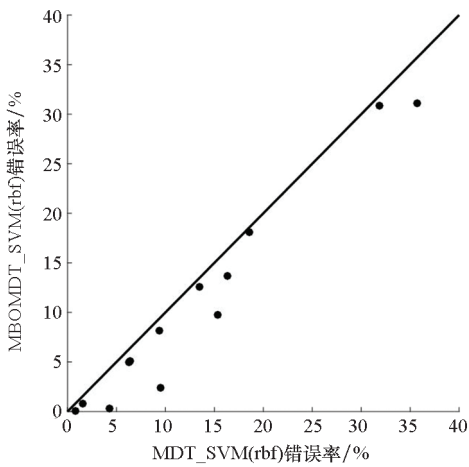
MBOMDT 算法只有在其中1个数据集上的结果与 MDT 算法基本持平,在剩下的数据集上,表现均有明显提升。特别地,在 Diabetis 数据集上 MBOMDT_SVM(rbf)算法的分类错误率比 MDT_SVM(rbf)算法降低了91%。综上,MBOMDT 算法可通过找到全局最优超参数使模型分类的性能得到进一步提升。



(a) LIBLINEAR



(b) SVM(linear)



(c) SVM(rbf)

图 3 MBOMDT 与 MDT 分类错误率比较

Fig. 3 Comparison of classification error rate for MBOMDT and MDT

2.4 本文算法与 BOMDT 算法的比较

本节将本文所提出的三种算法 MBOMDT_LIB、MBOMDT_SVM(linear)以及 MBOMDT_SVM(rbf)与使用原始贝叶斯优化后的模型决策树 BOMDT_LIB、BOMDT_SVM(linear)和 BOMDT_SVM(rbf)三种算法进行分类性能的比较。各算法均在最优超参数下运行,结果如表 6 所示。

由表 6 可知,MBOMDT_SVM(rbf)算法表现最好,表中标黑数据为该数据集下 6 种算法分类错误率最小值。从表中可以看出,三种 MBOMDT 算法在绝大部分数据集上的分类性能均优于 BOMDT 算法,这说明使用三种不同类型的核函数作为模型协方差函数组合要优于单一核函数下的贝叶斯优化算法,实验结果表明,本文所提算法可应对不同分类数据的特性,从而获得了较为理想的模型预测性能。

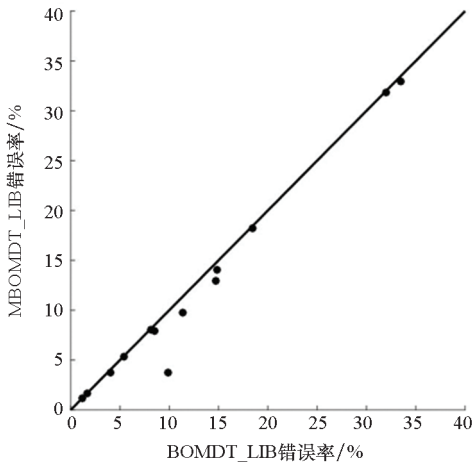
图 4 给出了 MBOMDT 算法与 BOMDT 算法采用三种不同分类器时分类错误率的大小关系比较。由图 4 可知,只有采用 SVM(linear)模型时,在 1 个数据集上略低于 BOMDT 算法,其他算法均好于或与 BOMDT 算法相当。特别地,在 Image 数据集上 MBOMDT_SVM(rbf)算法的分类错误率比 BOMDT_SVM(rbf)算法降低了 73%。综上,MBOMDT 算法通过在高斯过程中构建三个核函数组合,对分类问题适用更加广泛,且在一定程度上可以提升模型的性能。

表 6 不同算法下分类错误率比较结果

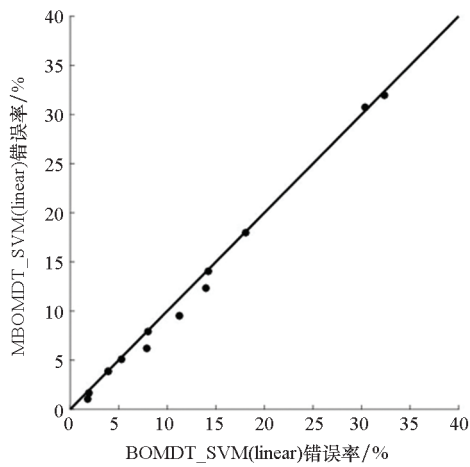
Tab. 6 Classification error rates comparison results of different algorithms

数据集	BOMDT			MBOMDT		
	LIBLINEAR	SVM(linear)	SVM(rbf)	LIBLINEAR	SVM(linear)	SVM(rbf)
Australian	14.670	14.000	13.670 0	13.000	12.330	13.660 0
Madelon	33.500	30.350	32.550 0	33.000	30.700	31.150 0
Germes	8.570	7.933	6.000 0	7.900	6.143	5.143 0
Breast_cancer	1.688	1.948	2.468 0	1.688	1.688	2.338 0
Spambase	8.133	8.000	15.530 0	8.000	7.900	9.733 0
Diabetis	4.033	3.967	1.167 0	3.733	3.833	0.366 7
Splice	1.177	1.821	0.947 1	1.177	1.039	0.855 2
Image	9.901	3.861	0.297 0	3.800	3.861	0.079 2
Banana	11.400	11.200	8.400 0	9.800	9.500	8.100 0
CCC	18.440	18.100	18.140 0	18.200	18.000	18.070 0
MGT	14.820	14.250	13.010 0	14.100	14.000	12.580 0
Cod_rna	5.400	5.220	5.390 0	5.300	5.100	4.915 0
Flare_solar	32.020	32.420	30.840 0	31.900	31.920	30.840 0

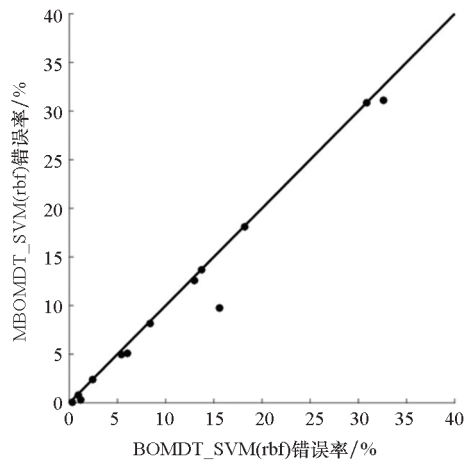
%



(a) LIBLINEAR



(b) SVM(linear)



(c) SVM(rbf)

图4 MBOMDT与BOMDT分类错误率比较

Fig.4 Comparison of classification error rate for MBOMDT and BOMDT

3 结论

本文提出一种多核贝叶斯优化的模型决策树算法,算法采用三种不同的高斯过程建模寻优,以应对不同的分类数据特性,利用贝叶斯优化技术,

不断加入新的信息,找到新的后验分布,选出最佳的参数组合。实验表明,所提算法在参数寻优上要优于传统的MDT寻优方法,能够在迭代次数不多的情况下找到更优的参数值,且在一定程度上提升算法的分类性能,减少计算代价。不过多核贝叶斯优化的模型决策树算法对数据维度和算力要求方面比较高,在未来的工作中,需研究数据降维对算法的影响,以增强本算法的适用性。

参考文献 (References)

- [1] GIANAZZA D. Forecasting workload and airspace configuration with neural networks and tree search methods[J]. *Artificial Intelligence*, 2010, 174(7/8): 530–549.
- [2] 王文剑, 门昌骞. 支持向量机建模及应用[M]. 北京: 科学出版社, 2014.
WANG W J, MEN C Q. Support vector machine modeling and application [M]. Beijing: Science Press, 2014. (in Chinese)
- [3] BREIMAN L. Random forests[J]. *Machine Learning*, 2001, 45(1): 5–32.
- [4] CHEN T Q, GUESTRIN C. XGBoost: a scalable tree boosting system[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016: 785–794.
- [5] KE G L, MENG Q, THOMAS F, et al. LightGBM: a highly efficient gradient boosting decision tree[C]// Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017.
- [6] FALKNER S, KLEIN A, HUTTER F. BOHB: robust and efficient hyperparameter optimization at scale [C]// Proceedings of the 35th International Conference on Machine Learning, 2018.
- [7] NARAYAN S, KRISHNA C S, MISHRA V, et al. Ultron-AutoML: an open-source, distributed, scalable framework for efficient hyper-parameter optimization [C]//Proceedings of IEEE International Conference on Big Data (Big Data), 2020: 1584–1593.
- [8] LIU J Y, TRIPATHI S, KURUP U, et al. Auptimizer—an extensible, open-source framework for hyperparameter tuning[C]//Proceedings of IEEE International Conference on Big Data (Big Data), 2019: 339–348.
- [9] SHAHRIARI B, SWERSKY K, WANG Z Y, et al. Taking the human out of the loop: a review of Bayesian optimization[J]. *Proceedings of the IEEE*, 2016, 104(1): 148–175.
- [10] KLEIN A, FALKNER S, BARTELS S, et al. Fast Bayesian optimization of machine learning hyperparameters on large datasets[EB/OL]. (2017-03-07) [2021-05-26]. <https://arxiv.org/abs/1605.07079v1>.
- [11] BROCHU E, CORA V M, DE FREITAS N. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning[EB/OL]. (2010-12-12) [2021-05-26]. <https://arxiv.org/abs/1012.2599>.
- [12] WANG Z Y, SHAKIBI B, JIN L, et al. Bayesian multi-scale optimistic optimization[EB/OL]. (2014-2-27) [2021-05-28]. <https://arxiv.org/abs/1402.7005>.
- [13] VANCHINATHAN H P, NIKOLIC I, DE BONA F, et al.

- Explore-exploit in top-N recommender systems via Gaussian processes [C]//Proceedings of the 8th ACM Conference on Recommender Systems, 2014: 225–232.
- [14] KHAJAH M M, ROADS B D, LINDSEY R V, et al. Designing engaging games using Bayesian optimization [C]//Proceedings of the CHI Conference on Human Factors in Computing Systems, 2016: 5571–5582.
- [15] GARNETT R, OSBORNE M A, ROBERTS S J. Bayesian optimization for sensor set selection [C]//Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, 2010: 209–219.
- [16] MARCHANT R, RAMOS F. Bayesian optimisation for intelligent environmental monitoring [C]//Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012: 2242–2249.
- [17] COLOPY G W, ROBERTS S J, CLIFTON D A. Bayesian optimization of personalized models for patient vital-sign monitoring [J]. IEEE Journal of Biomedical and Health Informatics, 2018, 22(2): 301–310.
- [18] HUTTER F, HOOS H H, LEYTON-BROWN K. Sequential model-based optimization for general algorithm configuration [C]//Proceedings of the International Conference on Learning and Intelligent Optimization, 2011: 507–523.
- [19] BERGSTRA J, BENGIO Y. Random search for hyperparameter optimization [J]. Journal of Machine Learning Research, 2012, 13(1): 281–305.
- [20] 崔佳旭, 杨博. 贝叶斯优化方法和应用综述 [J]. 软件学报, 2018, 29(10): 3068–3090.
- CUI J X, YANG B. Survey on Bayesian optimization methodology and applications [J]. Journal of Software, 2018, 29(10): 3068–3090. (in Chinese)
- [21] SRINIVAS N, KRAUSE A, KAKADE S M, et al. Gaussian process optimization in the bandit setting: no regret and experimental design [EB/OL]. (2010–01–09) [2021–05–28]. <https://arxiv.org/abs/0912.3995v4>.
- [22] DESAUTELS T, KRAUSE A, BURDICK J. Parallelizing exploration-exploitation tradeoffs with Gaussian process bandit optimization [EB/OL]. (2012–01–27) [2021–05–28]. <https://arxiv.org/abs/1206.6402>.
- [23] 肖跃雷, 张云娇. 基于特征选择和超参数优化的恐怖袭击组织预测方法 [J]. 计算机应用, 2020, 40(8): 2262–2267.
- XIAO Y L, ZHANG Y J. Terrorist attack organization prediction method based on feature selection and hyperparameter optimization [J]. Journal of Computer Applications, 2020, 40(8): 2262–2267. (in Chinese)
- [24] 尹儒, 门昌骞, 王文剑, 等. 模型决策树: 一种决策树加速算法 [J]. 模式识别与人工智能, 2018, 31(7): 643–652.
- YIN R, MEN C Q, WANG W J, et al. Model decision tree: an accelerated algorithm of decision tree [J]. Pattern Recognition and Artificial Intelligence, 2018, 31(7): 643–652. (in Chinese)
- [25] RAISSI M, BABAEE H, KARNIADAKIS G E. Parametric Gaussian process regression for big data [J]. Computational Mechanics, 2019, 64: 409–416.
- [26] YANG K F, EMMERICH M, DEUTZ A, et al. Efficient computation of expected hypervolume improvement using box decomposition algorithms [J]. Journal of Global Optimization, 2019, 75: 3–34.
- [27] QIN C, KLABJAN D, RUSSO D. Improving the expected improvement algorithm [C]//Proceedings of the 31st Conference on Neural Information Processing Systems, 2017.
- [28] CONTAL E, BUFFONI D, ROBICQUET A, et al. Parallel Gaussian process optimization with upper confidence bound and pure exploration [C]//Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2013: 225–240.
- [29] 王文剑, 田萌. 核选择研究进展 [J]. 山西大学学报(自然科学版), 2017, 40(3): 460–471.
- WANG W J, TIAN M. Advances in kernel selection research [J]. Journal of Shanxi University (Natural Science Edition), 2017, 40(3): 460–471. (in Chinese)