方法, 按照一定的权重集成 LR、XGBoost、LightGBM, 形成 BLLX 模型. 然后使用 Stacking 思想, 分别集成 XGBoost 和 LightGBM, 形成 SXL 模型.

2 集成模型

本节分析研究所使用的集成模型以及基础模型, 第一部分介绍基础模型, 第二部分介绍集成模型, 第三部分分析模型的优缺点.

2.1 基础模型

2.1.1 LR

LR 模型是优秀的二分类预测模型之一, 常常被众多研究学者使用在点击率的预测问题中. 对于二分类任务来说, 它的输出一般只有两种情况: 0 或者 1. LR 模型将线性函数的结果通过 Sigmoid 函数计算, 输出最终的预测结果. 其基本模型如式 (1) 所示. 其中 Y 为预测结果, x 为样本, w 为样本特征的权重向量, b 为模型的偏置. LR 模型一般分为 3 部分解决点击率预测问题: 第一, 寻找模型的预测函数 $f(x)$, 通常我们使用 sigmoid 函数作为预测函数 $f(x)$; 第二, 建立损失函数 $loss(w)$, 一般使用极大似然估计法建立 $loss(w)$; 第三, 求解使损失函数 $loss(w)$ 最小的参数 w, 一般使用梯度下降的方法求解 w.

$$Y = \frac{1}{1 + e^{-(w^{\mathrm{T}}x+b)}} \tag{1}$$

2.1.2 XGBoost

XGBoost[16]是大规模并行 boosted tree 的工具, 它是目前最快最好的开源 boosted tree 工具包, 比常见的工具包快 10 倍以上. XGBoost 是以分类回归树 CART 为基础, 对多个 CART 进行组合. 对于单个的 CART, 需要找到损失函数最小的一个分类回归树, 而 XGBoost 通过加法模型来组合多个 CART, 将模型上次预测 (由 t–1 颗树组合成的模型) 产生的误差作为参考下一棵树 (第 t 颗树) 的建立. 因此, 每加入一棵树, 将其损失函数不断降低. 其算法的流程如下所示:

1) 在每次迭代过程中加入一颗新树 $f(x_i)$;

2) 在每次迭代计算 $f(x_i)$ 的已一阶导数和二阶导数;

3) 计算 $f(x_i)$ 的目标函数的最小损失值 Obj, 并根据值 Obj 来生成树 $f(x_i)$.

2.1.3 LightGBM

尽管很多学者使用不同的优化算法来提高 GBDT 的效率, 但当数据特征维度过高、数据量过大时, 这些算法总是那么不尽人意. 他们共同不足的地方是, 在计算信息增益时都需要扫描所有的样本, 来找到最佳的划分点, 从而消耗了大量的计算时间. 因此, 微软为了解决这方面的问题, 提出了 Gradient-based One-Side Sampling 梯度单边采样 (GOSS) 和 Exclusive Feature Bundling 互斥特征绑定 (EFB) 两个算法优化 GDBT, 将优化后的 GDBT 称为 LightGBM[17].

GOSS[17]提出是为了证明梯度较大的样本在计算信息增益的时起重要作用, 从而能够从数量较小的样本中获得相当准确的信息增益估计值. 其算法的核心思想是: 在总样本中选取梯度较大的部分样本, 并在剩下样本中随机选取出部分样本, 两者组合成新的样本来学习新的分类器. 这样的做法是为了采样的样本与总样本的分布一致和训练小梯度样本数据, 从而在不改变样本的分布前提下不损失分类器学习的精确并且大大的减少了分类器学习的速率.

EFB[17]是一种能够减少高维数据的特征数并使损失最小的一种算法, 将稀疏特征空间中的非 0 值的特征绑定到一起形成一个特征, 然后从特征绑定中建立相同的特征直方图作为单一特征, 通过这种方式能够在无损精度的情况加速 GBDT 的训练.

2.2 基于学习法和平均法的集成模型

本文基于 LR、XGBoost 和 LightGBM3 种单一模型, 使用 Stacking 和 Blending 集成思想, 提出 2 种集成模型: SXL 和 SSL.

2.2.1 SXL 模型

SXL 模型通过 Stacking 技术, 集成 LightGBM 和 XGBoost 两种单一模型, 其整体结构如图 1 所示. SXL 中 S 代表集成模型的技术 Stacking, X 和 L 分别代表集成模型的基础模型 XGBoost 和 LightGBM. Stacking 初等模型层使用 XGBoost 模型进行 5 折交叉验证, 最终模型从使用 LightGBM 作预测. 5 折交叉验证是将原始数据分成 5 等分, 每次训练选四折作为训练数据, 另外一折作为测试数据, 每一折训练得到一个预测结果, 循环五次, 最后将五次预测结果和原始数据拼接, 得到最终模型从的训练数据. 最终模型使用 LightGBM 训练初等模型层得到的数据, 输出最终的预测结果.

SXL 模型一共分为两层, 第一层为初等模型层, 其中选用 XGBoost 为基础训练模型; 第二层为最终模型层, 选用 LightGBM 为基础训练模型. SXL 模型的实现

主要分为以下几个步骤: 1) 首先读取原始特征数据集 I; 2) 按原始特征数据集 I, 平均分成 5 等份 I_i (i∈1, 2, 3, 4, 5); 3) 对每一个训练 XGBoost 模型, 得到 $Result_i$; 4) 组合原始特征数据集 I 和初等模型层训练后得到的 $Result_i$ 组合成新特征集 D; 5) 用训练最终模型层 LightGBM, 得到最终预测结果 $Result$.

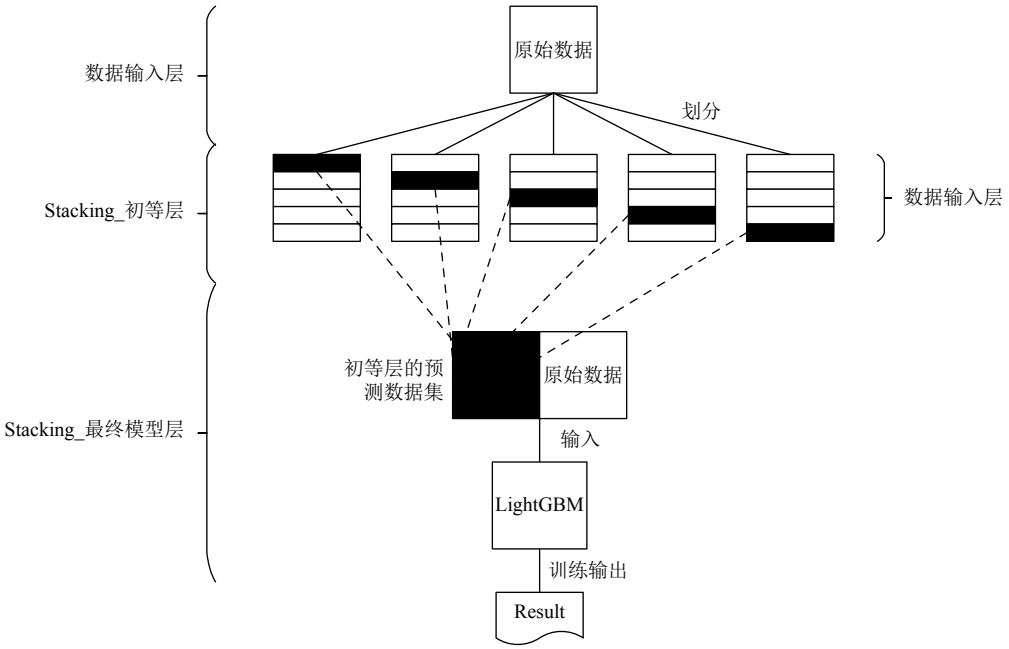

图 1 SXL 集成模型图

2.2.2 BLLX 模型

BLLX 模型使用 Blending 技术, 集成 LR、LightGBM、XGBoost 3 种基础单一模型. 整体结构如图 2 所示. BLLX 模型是集成 3 种单一模型的分别训练后的结果, 通过一定的权重分配, 来获得最后的结果. 经过实验结果的对比, 我们最终选择的模型权重如式 (2).

$$BLLX = 0.25 * result_1 + 0.25 * result_2 + 0.5 * result_3 \quad (2)$$

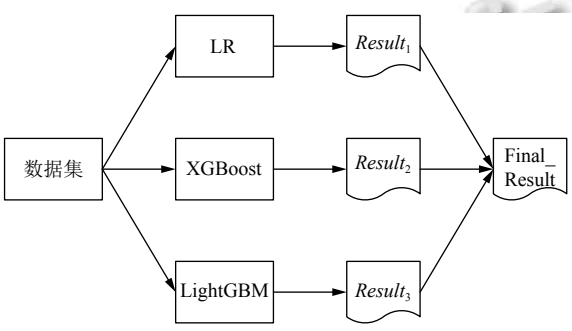

图 2 BLLX 集成模型图

2.3 性能分析

对于分析数据集, 不同的方法处理数据集获得的结论不同, 不同的机器学习模型学习数据集的方法不同, 其训练结果也不一样. 集成技术能够训练单一模型, 结合它们的优点, 提高预测的能力. XGBoost 改进了 GBDT 的残差函数, 利用 CPU 的多线程, 引入正则化项, 控制模型的复杂度, 使用预排序的方法实现特征并行. LightGBM 使用直方图算法, 使训练过程加速, 拥有更高的训练效率, 相比于 XGBoost, 它占用更低的内存, 具备处理大数据的能力. SXL 模型使用 Stacking 集成思想, 集成了 XGBoost 和 LightGBM 的优点, 既能够并行进行训练, 又能够得到更准确的预测结果. 但是模型训练中多次训练单一模型, 重复读取大量的数据, 会导致模型的训练时间加倍增长, 影响模型的效率, 而且 Stacking 技术增加了整个模型的复杂度, 容易训练过拟合. BLLX 模型是在 LR、XGBoost 和 LightGBM 的各自训练后的结论上进行加权平均, 不会导致过拟合的发生, 其模型复杂度也没有 SXL 模型高, 更容易得到不错的预测结果. 但是如何处理各个模型的权重的问题上, 需要经过大量的时间计算, 分配每个单一模型的权重值.

3 实验结果及分析

3.1 实验数据与预处理

本文采用的数据集来自腾讯社交广告高校算法大

赛, 该项比赛是以移动 App 广告为研究对象, 要求比赛参与者利用腾讯社交广告平台中的真实数据预测 App 广告转化率, 从而能够提高广告的投放效果, 并且扩大广告带来的相关收入. 原始数据中包含了以下数据特征: 1) 广告相关特征 (广告主信息、广告的相关信息等); 2) 用户特征 (用户相关信息年龄、性别、学历、教育情况、婚姻情况等等); 3) 上下文特征 (用户使用的手机类型、运营商、联网方式和广告位置的相关信息等).

由于数据来源于真实的广告平台中, 每天都能产生大量的历史数据, 每位用户都可能产生大量的广告日志记录, 会造成数据中用户与转化成功的广告数量比严重不足. 且在数据采集过程中, 由于一些不可控的原因, 会导致数据集中的数据缺失问题. 针对此问题, 我们根据缺失数据所属的特征进行分析, 若它为连续型特征, 我们会根据它的均值进行补充缺失值, 若它为离散型特征, 则我们会删除该条数据, 虽然删除缺失值会影响后续模型预测的准确性, 但是在本次的 App 广告数据集中, 缺失的样本几乎只占了样本总量的 0.1%, 删除它并不会影响样本的总体的分布. 在经过处理后, 我们的数据样本中包含 9386 404 条数据, 其中转化的 App 广告条数为 234 382 条, 未转化的 App 广告条数为 9152 022 条.

3.2 评估标准

本文研究的问题是针对数据集中的每条数据样本, 预测用户点击并激活 App 广告的转化率. 其本质上属于一个二分类问题, 因为用户是否激活该 App 广告, 只有两个取值: 0 或者 1. 对数损失 (Log-likelihood Loss, *LogLoss*) 是在概率论上定义的一个评估分类器的概率输出的标准. 它能够通过惩罚错误的分类, 完成对评估分类器的准确度的量化, 为了计算 *LogLoss* 值, 每个分类器必须提供样本的所属类别的概率值. 对于一个二分类问题, *LogLoss* 的计算公式如式 (3) 所示:

$$LogLoss = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log p_i + (1-y_i)\log(1-p_i)) \quad (3)$$

式中, y_i 代表第 i 个样本 x_i 的类别, 取值为 0 或 1, p_i 代表分类器输出的样本 x_i 的预测概率值, 取值区间在 0 到 1, N 为样本总量.

3.3 特征提取

3.3.1 基础特征

基础特征为数据初始的特征, 训练集中的每一个原始字段都可以作为基础特征, 从而统计相关统计量 (例如过去几天内的用户点击总量、用户是否转化过当前 App 等). 以训练集中的素材 creative_id 为例, 它是用户最直接看到的内容, 对某个特定的 creative_id, 我们统计过去若干天内该 creative_id 的总点击量、转化次数和转化率, 作为该 creative_id 取值的 3 个特征. 它的物理意义是量化地描述了该素材是否更能吸引用户 (点击量) 和发生转化 (转化率).

3.3.2 用户信息特征

基础特征以外, 用户信息特征是非常重要的特征. 因为一个 App 广告是否得到激活转化, 都是由用户主观上决定的, 所以针对用户的相关行为进行分析对本次实验是十分重要的. 我们通过对用户行为的分析, 提取了一些相关的特征, 包括: 用户安装 App 的总量、用户在转化该广告之间的一些行为 (点击之前用户点击 App 的数量等)、用户点击相同素材的广告总量等等一些与用户相关的行为特征.

3.3.3 贝叶斯平滑特征

贝叶斯平滑假设所有的广告都有一个自身的转化率, 这些转化率服从于一个 Beta 分布, 其次对于某一个广告, 给定转化次数和它自身的转化率, 它的点击次数服从一个伯努利分布, 最后用梯度下降来学习这个分布. 当我们预测 App 广告 CVR 时, 机器学习模型非常依赖于统计特征, 每个广告的反馈 CVR 都能够极大的提升预测的准确性. 我们使用历史数据来获得 App 广告的 CVR 时存在一个问题, 即在特种提取中我们统计了同一广告位 App 的历史转化率, 由于广告位上线有前后区别, 而且上线慢的广告位统计不充分, 大多数用户只点击过 App 广告一次, 那么它的历史转化率就是 100%. 如果拿这个特征训练模型, 可能导致数据偏差. 在贝叶斯平滑中, 我们一共平滑了两种数据, 一种是 App 广告位置信息的贝叶斯平滑率 (CF_pos), 一种是 App 广告素材信息的贝叶斯平滑率 (CF_cre).

3.4 实验结果

3.4.1 模型参数设置

在设置 XGBoost 和 LightGBM 模型参数时, 我们选择使用 Python 学习库中的 GridSearchCV 方法, 对模型进行参数进行交叉验证选择出模型的合适参数, 其参数设置如表 1 和表 2 所示.

3.4.2 特征提取的影响

我们使用 GBDT 模型中对特征评估的方法, 对我

们在特征提取阶段得到的特征进行评估, 得到如图 3 所示的重要性得分图. 图中以 SF 开始的特征为用户信息特征, CF 开始的特征为贝叶斯特征, 小写字母开始和 BF 开始的特征为基础特征. 从图中我们可以得到, 在经过贝叶斯平滑后得到的两个特征 CF_pos 和 CF_cre 的重要性得分最高, 我们可以判断出 App 广告的位置和广告使用的素材对 App 广告成功转化有较大的影响. 在基础特征中, 年龄 age 对 App 广告成功转化的影响最大, appplatform 得分最低, 我们可以得到年龄是决定 App 广告转化的重要因素之一, 而 App 的平台 (安卓和苹果) 对 App 广告成功转化有微弱的影响. 因此我们筛选出得分少于 100 的特征, 得到最终的特征集.

表 1 XGBoost 参数表

参数	值
Objective	binary:logistic
learning_rate	0.10
n_estimators	500
max_depth	7
min_child_weight	4
gamma	0.4
subsample	0.8
colsample_bytree	0.8
reg_lambda	6

表 2 LightGBM 参数表

参数	值
boosting_type	gbdt
objective	binary
metric	binary_logloss
num_leaves	40
max_depth	7
max_bin	20
min_data_in_leaf	1
feature_fraction	0.6
bagging_fraction	0.3
bagging_freq	10
reg_alpha	0.2
reg_lambda	0.01
min_split_gain	0.8
n_estimators	700
learning_rate	0.01

接着, 我们使用 XGBoost 和 lightGBM 两种单一模型对特征进行训练. V1 阶段我们使用原始数据集, V1-V2 和 V2-V3 我们加入基础特征, V3-V4、V4-V5 和 V5-V6 阶段我们加入用户信息特征, V6-V7 阶段我们加入贝叶斯平滑特征. 从图 4 中我们可以看出, 随着训练集中数据特征的增多, 两个模型的 *LogLoss* 值趋向于下降, 这证明我们提取的特征能够大幅度的提升模型的预测效果.

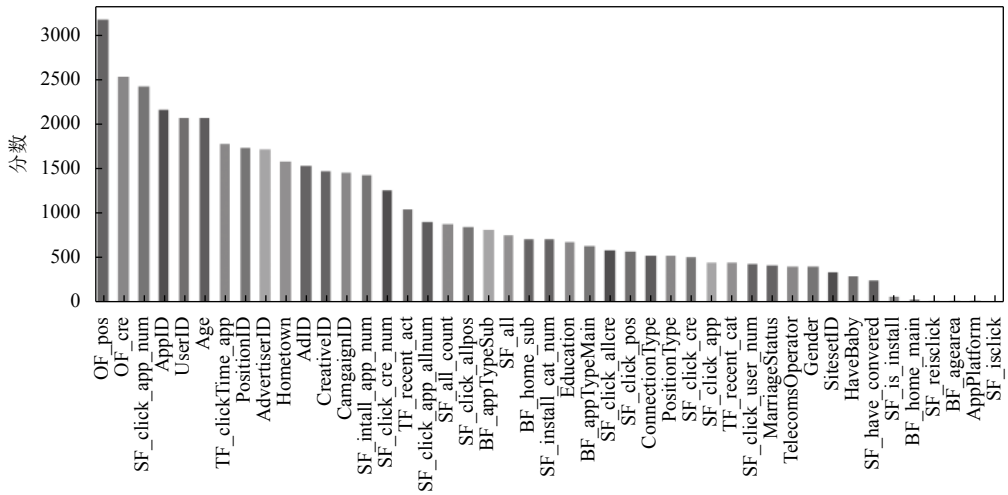

图 3 特征重要性得分图

3.4.3 预测结果分析

在本文实验中, 一共训练了 7 种模型: LR、XGBoost、LightGBM3 种单一模型, GBDT+LR、RF+LXFV 两种用于广告转化率预测的集成模型, 以及本文提出了两种集成模型 SXL 和 BLLX, 7 种模型的训练结果如表 3 所示. 从表中可以看出, LR 模型的效果最差, 其 *LogLoss* 为 0.1033; BLLX 模型的效果最好其 *LogLoss* 为 0.0922; 在单一模型中, XGBoost 和 LightGBM 的预测能力均比 LR 模型优秀; 而集成模型与单一模型对比中, GBDT+LR 稍微比单一模型差, 而其它 3 种集成模型均比单一模型好; 在 4 种集成模型对比中, BLLX 模型和 SXL 模型的效果比另外 2 种好; 在时间成本上, RF+LXFV 的

时间成本最高, 其原因在于集成了 4 种单一模型, 导致模型的复杂度提升, 从而导致训练时间过长; SXL 模型在集成模型中, 训练时间最短, 为 59.8 分钟.

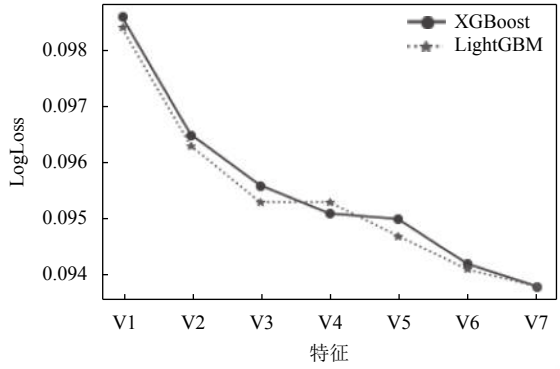

图 4 不同特征在模型上的效果图

表 3 各模型 App 广告转化预测的 *LogLoss* 和时间成本

模型	*LogLoss*	Time(min)
LR	0.1033	15.3
LightGBM	0.0938	21.3
XGBoost	0.0938	35.5
GBDT+LR	0.0945	60.9
RF+LXFV	0.0936	110.8
SXL	0.0928	59.8
BLLX	0.0922	72.1

实验结果充分证明了我们提出的 SXL 模型和 BLLX 模型的有效性与预测转化率的精准度, 在实际使用中, BLLX 模型的时间成本比 SXL 模型高, 但 *LogLoss* 比 BLLX 模型低, 如果公司使用模型用在广告系统中, 考虑到预测转化率的精度微弱提高都会带来巨大的收益情况下, 建议使用 BLLX 模型; 而如果因为时间成本问题, 在损失预测精度的情况下, 可以考虑 SXL 模型.

4 结论

本文通过对 App 广告点击后激活的转化率预测问题的研究, 使用特征工程, 处理数据集, 并提出了两种集成模型 SXL 和 BLLX. 论文以腾讯社交广告算法大赛中的实际数据为基础, 通过特征工程挖掘出大量的用户数据特征, 提供了模型训练的训练集, 为模型训练打下了坚实的基础. 集成模型 SXL 和 BLLX 在 *LogLoss* 方法评估中都要明显高于传统机器学习模型和其它集成模型, 本文模型的有效性得到了充分的验证. 当然本文也存在很多不足的地方, 例如在特征提取中, 没有考虑到时间信息对 App 广告转化率的影响. 在今后的研究中, 考虑挖掘时间信息, 进一步提高模型的预测能力.

参考文献

1 Alaimo C, Kallinikos J. Objects, metrics and practices: An inquiry into the programmatic advertising ecosystem. Proceedings of IFIP WG 8.2 Working Conference on the Interaction of Information Systems and the Organization. San Francisco, CA, USA. 2018. 110–123.

2 Fain DC, Pedersen JO. Sponsored search: A brief history. Bulletin of the American Society for Information Science and Technology, 2006, 32(2): 12–13. [doi: 10.1002/bult.1720320206]

3 Graepel T, Candela JQ, Borchert T, *et al*. Web-scale bayesian click-through rate prediction for sponsored search advertising in Microsoft's Bing search engine. Proceedings of the 27th International Conference on Machine Learning. Haifa, Israel. 2010.

4 Jansen BJ, Mullen T. Sponsored search: An overview of the concept, history, and technology. International Journal of Electronic Business, 2008, 6(2): 114–131. [doi: 10.1504/IJEB.2008.018068]

5 Goldberg DE, Holland JH. Genetic algorithms and machine learning. Machine Learning, 1988, 3(2-3): 95–99. [doi: 10.1007/BF00113892]

6 潘博, 张青川, 于重重, 等. FM 集成模型在广告点击率预估中的应用. 计算机应用与软件, 2018, 35(1): 107–111, 148.

7 Richardson M, Dominowska E, Ragno R. Predicting clicks: Estimating the click-through rate for new ads. Proceedings of the 16th International Conference on World Wide Web. Banff, Canada. 2007. 521–530.

8 Shan LL, Lin L, Sun CJ, *et al*. Predicting ad click-through rates via feature-based fully coupled interaction tensor factorization. Electronic Commerce Research and Applications, 2016, 16: 30–42. [doi: 10.1016/j.elerap.2016.01.004]

9 Joachims T. Optimizing search engines using clickthrough data. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton, AB, Canada. 2002. 133–142.

10 He XR, Pan JF, Jin O, *et al*. Practical lessons from predicting clicks on ads at facebook. Proceedings of the 8th International Workshop on Data Mining for Online Advertising. New York, NY, USA. 2014. 1–9.

11 Liu H, Motoda H. Feature Extraction, Construction and Selection: A Data Mining Perspective. Boston: Springer,

1998.
12 Regelson M, Fain DC. Predicting click-through rate using keyword clusters. Proceedings of the 2nd Workshop on Sponsored Search Auctions. MI, USA. 2006. 1–6.
13 周志华. 机器学习. 北京: 清华大学出版社, 2016.
14 Ling XL, Deng WW, Gu C, *et al*. Model ensemble for click prediction in bing search ads. Proceedings of the 26th International Conference on World Wide Web Companion. Perth, Austrilia. 2017. 689–698.
15 赵杨, 袁析妮, 陈亚文, 等. 基于机器学习混合算法的APP广告转化率预测研究. 数据分析与知识发现, 2018, 2(11): 2–9. [doi: 10.11925/infotech.2096-3467.2018.0834]
16 Chen TQ, Guestrin C. Xgboost: A scalable tree boosting system. Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining. San Francisco, CA, USA. 2016. 785–794.
17 Ke GL, Meng Q, Finley T, *et al*. LightGBM: A highly efficient gradient boosting decision tree. Proceedings of the 31st Conference on Neural Information Processing Systems. Long Beach, CA, USA. 2017. 3146–3154.