Computer Science and Application 计算机科学与应用, 2025, 15(5), 680-689
Published Online May 2025 in Hans. https://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2025.155140

Hans汉斯

# 基于采样增强与动态直方图的改进LightGBM算法

**张　林，严　涛**

南京理工大学数学与统计学院，江苏　南京

收稿日期：2025年4月22日；录用日期：2025年5月21日；发布日期：2025年5月28日

## 摘　要

**梯度提升类算法面临的主要问题是大规模数据下的运算速度问题。本文针对LightGBM中采样仅依赖一阶导数影响精度，以及直方图分箱忽视数据分布特征导致计算冗余，提出了基于牛顿法的梯度单边采样，引入二阶导数提高采样精度，同时设计动态直方图算法，实现分布和标签感知的自适应分箱。在Epsilon和MNIST8M数据集上的实验表明，新方法在提升模型性能的同时，训练时间分别减少了20.7%和9.8%。**

## 关键词

**LightGBM算法，采样方法，直方图算法**

# An Improved LightGBM Algorithm Based on Sampling Enhancement and Dynamic Histogram

**Lin Zhang, Tao Yan**

School of Mathematics and Statistics, Nanjing University of Science and Technology, Nanjing Jiangsu

Received: Apr. 22nd, 2025; accepted: May 21st, 2025; published: May 28th, 2025

## Abstract

**Gradient boosting algorithms face computational efficiency challenges when processing large-scale data. In order to improve the limitations in LightGBM: the gradient-based one-side sampling relying solely on first-order derivatives which compromises accuracy, and histogram binning ignoring data distribution characteristics leading to computational redundancy, we propose a Newton-based gradient one-side sampling method incorporating second-order derivatives to enhance precision,**

文章引用: 张林, 严涛. 基于采样增强与动态直方图的改进 LightGBM 算法[J]. 计算机科学与应用, 2025, 15(5): 680-689.
DOI: 10.12677/csa.2025.155140

**along with a dynamic histogram algorithm enabling distribution-aware and label-aware adaptive binning. Experimental results on the Epsilon and MNIST8M datasets demonstrate that our approach improves model performance while reducing training time by 20.7% and 9.8% respectively.**

## Keywords

**LightGBM Algorithm, Sampling Method, Histogram Algorithm**

Copyright © 2025 by author(s) and Hans Publishers Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/
Open Access

## 1. 引言

LightGBM [1]针对传统 GBDT [2]在大规模数据下的效率问题，提出了两项关键改进：基于梯度的单边采样(Gradient-Based One-Side Sampling, GOSS)和互斥特征捆绑(Exclusive Feature Bundling, EFB)。GOSS 保留部分大梯度样本并随机采样小梯度样本，在保证精度的同时降低计算复杂度。EFB 利用高维特征空间的互斥性，通过图着色算法将互斥特征捆绑，减少特征数量。LightGBM 还采用直方图算法和叶子优先策略，显著提升了训练效率和内存利用率。

依靠其高效性能和卓越的预测能力，LightGBM 已在多个领域获得广泛应用。在金融领域，Ponsam 等人[3]将其用于信用风险评估，显著提高了高风险客户识别率；Ge 等人[4]结合序列特征工程构建了高效的欺诈检测系统。医疗领域，Han 等人[5]利用 LightGBM 和欧几里得距离图改进了阿尔茨海默症的检测效果。自然语言处理方面，Alzamzami 等人[6]基于情感多媒体数据集实现了有效的情感分类。

近年来，研究者对其进行了多方面的改进和优化。Ong 等人[7]提出自适应直方图方法，根据任务类型调整分辨率，有效提升了不平衡数据集的处理能力；Zhang 等人[8]改进了 GPU 加速方案，通过优化特征直方图构建使训练速度提升 7~8 倍；Meng 等人[9]设计并行投票决策树算法，显著降低了大数据环境下的通信成本；Shi 等人[10]引入量化技术，在保证精度的同时降低了计算和通信开销。

本文提出了两种优化 LightGBM 的方法：基于牛顿法的梯度单边采样和基于分布感知与标签感知的动态直方图算法。前者利用二阶导数信息精准筛选样本；后者通过分布和标签感知，实现了自适应动态分箱。这些改进有效降低了计算复杂度并提升了模型的性能。

## 2. 算法优化

### 2.1. 基于牛顿法的梯度单边采样

在 LightGBM [11]中，损失函数定义为如下形式

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t\left(x_i\right)\right) + \Omega\left(f_t\right). \tag{1}$$

对于(1)，使用二阶展开近似的方式可以得到如下格式：

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n}\left[l\left(y_i, \hat{y}^{(t-1)}\right) + g_i f_t\left(\mathbf{x}_i\right) + \frac{1}{2} h_i f_t^2\left(\mathbf{x}_i\right)\right] + \Omega\left(f_t\right), \tag{2}$$

其中 $g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ 和 $h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \hat{y}^{(t-1)}\right)$ 分别为损失函数关于 $\hat{y}^{(t-1)}$ 的一阶导数和二阶导数。$l\left(y_i, \hat{y}^{(t-1)}\right)$ 为常数项，将常数项移除后，可以得到

$$\tilde{\mathcal{L}}^{(t)}=\sum_{i=1}^{n}\left[g_i f_t(\mathbf{x}_i)+\frac{1}{2}h_i f_t^2(\mathbf{x}_i)\right]+\Omega(f_t). \tag{3}$$

LightGBM 通过考虑一阶导数信息，执行基于梯度的单边采样，使用较少的样本提升了模型速度并维持了精度。但根据式(3)，损失函数的下降不仅和一阶导数相关，而且和二阶导数相关，因此在采样时同时考虑一阶导数信息和二阶导数信息，理论上能筛选出更值得训练的样本，进一步减少计算量和内存使用量。

从单个样本的角度考虑损失函数(3)，在不考虑正则项的情况下，对样本 $i$，可以得到单个样本的损失函数：

$$\tilde{\mathcal{L}}_i^{(t)}=g_i f_t(\mathbf{x}_i)+\frac{1}{2}h_i f_t^2(\mathbf{x}_i) \tag{4}$$

由于式(4)为关于 $f_t(\mathbf{x}_i)$ 的二次函数，则最小值点为 $f_t(\mathbf{x}_i)=-g_i/h_i$，将最小值点对应的值带入单个样本损失函数(4)，可得

$$\begin{aligned}\tilde{\mathcal{L}}_i^{(t)}&=g_i\left(-\frac{g_i}{hi}\right)+\frac{1}{2}h_i\left(-\frac{g_i}{h_i}\right)^2\\&=-\frac{g_i^2}{2h_i}\end{aligned}$$

取绝对值后，这一数值是当前样本在第 $t$ 次迭代可能下降的最大值，算法会构建决策树来满足这个下降的最大值，使得总体损失函数实现最大程度的下降，因此可以通过 $g_i^2/h_i$ 衡量样本对于损失函数下降的贡献，$g_i^2/h_i$ 较大，样本对于损失函数下降贡献大，反之则贡献小。定义 $g_i^2/h_i$ 为样本的重要性衡量指标。

基于 GOSS 方法的思想，我们希望通过采样筛选出最需要训练的样本，同时保持最终模型的性能基本不变，因此需要尽量保留重要的样本。具体而言，使用重要性指标 $g_i^2/h_i$ 代替一阶导数 $g_i$ 来衡量样本的重要性。在采样过程中：

(1) 重要样本保留：选取前 $a$% 的样本，这些样本的 $g_i^2/h_i$ 最高，以确保数据集的关键信息不丢失，损失函数可以获得最大的下降程度。

(2) 随机采样：对剩余样本进行 $b$% 的随机采样，减少计算量，同时使采样后样本的导数分布与采样前保持一致。

为了防止采样导致导数分布偏移，在计算信息增益时，需要对随机采样部分的一阶导数和二阶导数乘以一个调整因子 $(1-a)/b$，以补偿被丢弃的样本对导数分布的影响。这一调整能有效保持损失函数的下降量，从而保证训练效果的稳定性。图 1 是基于牛顿法的梯度单边采样示意图。

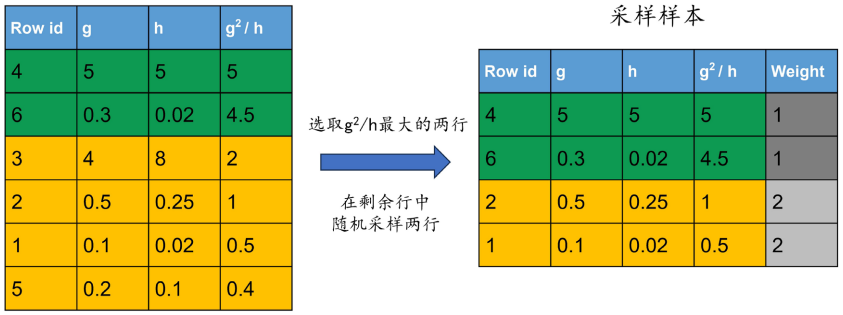


**Figure 1.** Newton-based gradient one-side sampling
**图 1.** 基于牛顿法的梯度单边采样

采样算法的伪代码见算法 1。

**算法 1** 基于牛顿法的单边采样

**Input:** $I$: 训练数据, $d$: 迭代次数, $loss$: 损失函数, $L$: 基学习器

**Input:** $a$: 高重要性样本的保留比例, $b$: 低重要性样本的采样比例

models $\leftarrow \{\}$, fact $\leftarrow \frac{1-a}{b}$, topN $\leftarrow a \times len(I)$, randN $\leftarrow b \times len(I)$

**for** $i = 1$ ***to*** $d$ **do**

  **if** $i == 1$ **then**

    models.append(initialModel)

    **continue** ▷ 跳到下一次循环

  preds $\leftarrow$ models.predict($I$)

  $g \leftarrow loss(I, preds)$, $h \leftarrow loss(I, preds)$, w $\leftarrow \{1, 1, ...\}$

  Calculate($\frac{g^2}{h}$)

  sorted $\leftarrow$ GetSortedIndices($\frac{g^2}{h}$)

  topSet $\leftarrow$ sorted[1:topN]

  randSet $\leftarrow$ RandomPick(sorted[topN:len($I$)], randN)

  usedSet $\leftarrow$ topSet + randSet

  w[randSet] $\times =$ fact

  newModel $\leftarrow$ L($I$[usedSet], $g$[usedSet], $h$[usedSet], w[usedSet])

  models.append(newModel)

## 2.2. 基于分布感知与标签感知的动态直方图算法

在 LightGBM 中，直方图分箱算法采用等频分箱策略，即每个箱体中包含大致相同数量的样本，这种方法在数据均匀分布的情况表现良好，但在处理以下几种情形的数据时会表现不佳：

(1) 数据分布不均匀，存在值域跨度较大的区间。

(2) 某些特定值出现频率较高，形成数据簇。

(3) 重要的决策边界可能位于数据稀疏区域。

这些情形下，单纯的等频分箱会导致较严重的信息损失，进而影响模型的整体性能。为了弥补等频分箱的不足，我们对其进行了改进，将相邻唯一值之间的差值(下文简称相邻差值)引入等频分箱，使得等频分箱可以对分布有所感知，改进后的分箱步骤如下：

**步骤一.** 对唯一值升序排序，计算每个唯一值对应的样本数量、唯一值总数、箱平均样本量和每个箱的平均差值，平均差值为极差除以样本总数，用于计算相邻差值的阈值。

**步骤二.** 按升序遍历唯一值，如果相邻差值和唯一值对应的样本数量同时超过阈值、唯一值对应的样本数量超过箱平均样本量或者累计唯一值数量达到箱平均样本量，进行分箱。

**步骤三.** 如果总箱数大于最大分箱数，依据相邻差值相近原则进行箱体合并，合并到最大分箱数为止。

进一步，对分类问题，引入基于标签信息的箱体合并，不同特征分箱数目依据标签信息确定，实现动态分箱，提升模型效率，箱体合并步骤如下：

**步骤一.** 等频分箱阶段将标签信息传入箱体。

**步骤二.** 对每个箱体而言，将不同类别的样本数量作为元素组成向量，依据这一向量对相邻两箱做卡方检验。

**步骤三.** 设定显著性水平，当 $p$ 值大于阈值时，认为两箱体分布无显著差异，可以合并，反之则不合

并。合并后将两箱的统计量相加。

**步骤四.** 迭代合并过程，直到所有相邻箱 $p$ 值都小于等于阈值或者达到预设最小分箱数。

结合分布感知和标签感知的直方图算法伪代码如算法 2 所示。

**算法 2** 基于分布感知与标签感知的动态直方图算法

**Input:** $I$: 训练数据, $depth$: 最大深度, $m$: 特征维度

**Input:** $span$: 相邻差值信息, $threshold$: 箱体合并阈值

```
nodeSet ← {0}: 当前层的节点
rowSet ← {{0, 1, 2, ...}} ▷ 节点的数据索引
for i = 1 to depth do
    for node in nodeSet do
        usedRows ← rowSet[node]
        for k = 1 to m do
            H ← new Histogram(span) ▷ 基于分布感知构建直方图
            for j in usedRows do
                bin ← I.f[j][k].bin ▷ 找到行号为 j，列号为 k 的样本分箱编号
                H[bin].n ← H[bin].n + 1
                H[bin].g ← H[bin].g + I.f[j][k].g ▷ 累计一阶导数
                H[bin].h ← H[bin].h + I.f[j][k].h ▷ 累计二阶导数
                H[bin].class1Num ←H[bin].class1Num + 1 累计第 1 个类别的数量
                H[bin].class2Num ←H[bin].class2Num + 1 累计第 2 个类别的数量
                ...
                H[bin].classnNum ←H[bin].classnNum + 1 累计第 n 个类别的数量
            MergeBins(threshold, H[bin].class1Num, H[bin].class2Num, ...) ▷ 合并箱体
            bestSplit=FindBestSplitInH(H) ▷ 在 H 中找最优分支点
    UpdateRowSet(rowSet, bestSplit) ▷ 根据最优分支点更新 rowSet
    UpdateNodeSet(nodeSet, bestSplit) ▷ 根据最优分支点更新 nodeSet
```

## 3. 数值实验

### 3.1. 实验设置

实验共使用 6 个数据集，数据集具体信息见表 1。Allstate (Allstate Insurance Claim)来源于 Allstate 保险公司，数据包含多种类别型和数值型特征，特征间存在复杂的非线性关系，并且特征中存在噪声。Flight Delay 来源于美国交通统计局，数据具有明显的时间序列特性，包含季节性和周期性模式。Higgs 是由蒙特卡罗法模拟生成的，类别相对平衡，为稠密数据集，所有特征均是数值型。Epsilon 由 Pascal 大规模学习挑战赛(Pascal Large Scale Learning Challenge)提供，旨在评估机器学习算法在高维特征空间中的性能，所有特征皆为数值型。MNIST8M 是一个扩展版的手写数字识别数据集，它是通过对原始 MNIST 数据集应用随机变形和平移生成的，数据部分稀疏，标签共 10 个类别。Istella LTR 是意大利搜索引擎公司 Istella 发布的，共包含 33018 个查询，总计 10454629 个查询文档对，这是一个高度稀疏的数据集，特征值分布较不平衡，存在长尾分布现象。这些数据集具有不同规模、不同稀疏性和不同分布，覆盖了实际应用的诸多场景，用这些数据集进行实验具有较好的代表性。

**Table 1.** Dataset information
**表 1.** 数据集信息

| 数据集 | 任务 | 数据量 | 特征数 | 备注 |
|---|---|---|---|---|
| Allstate | 二分类 | 13,184,290 | 4,220 | 稀疏数据 |
| Flight Delay | 二分类 | 10,100,000 | 700 | 稀疏数据 |
| Higgs | 二分类 | 10,500,000 | 28 | 稠密数据 |
| Epsilon | 二分类 | 500,000 | 2,000 | 稠密数据 |
| MNIST8M | 多分类 | 4,000,000 | 784 | 部分稀疏 |
| Istella LTR | 排序 | 10,454,629 | 220 | 稀疏数据 |

对二分类任务，实验选取的性能指标是 AUC；对多分类任务，实验选取的指标是准确率；对排序任务，实验选取的指标是 NDCG (Normalized Discounted Cumulative Gain)，截断位置为 10。

实验中线程数固定为 32，具体实验环境如表 2 所示。

**Table 2.** Experimental setup
**表 2.** 实验环境

| OS | CPU | CORE | MEMORY | THREAD |
|---|---|---|---|---|
| openEuler 22.03 | 2 * Intel(R) Xeon(R) Gold 6330 | 56 | 512 | 112 |

## 3.2. 实验结果分析

### (1) 综合比较

综合比较中，基于牛顿法的梯度单边采样被记为 NGOSS，采样改进后的 LightGBM 被记为 lgb_ngoss，同时使用采样改进和直方图改进的 LightGBM 被记为 lgb_enhanced，基线算法设定为使用直方图的 XGBoost [11] (xgb_hist)、不使用采样改进和直方图改进的 LightGBM [1] (lgb_baseline)、使用 GOSS 方法的 LightGBM [1] (lgb_goss)，综合比较将比较后五种算法。五种算法在测试集上的准确性如表 3 所示。

**Table 3.** Comparison of accuracy across different algorithms
**表 3.** 不同算法的准确性比较

| 数据集 | xgb_hist | lgb_baseline | lgb_goss | lgb_ngoss | lgb_enhanced |
|---|---|---|---|---|---|
| Allstate | 0.6078 | 0.6087 | 0.6086 | **0.6090 ± 5e-5** | **0.6090 ± 4e-5** |
| Flight Delay | 0.8501 | 0.8566 | 0.8564 | **0.8578 ± 7e-5** | 0.8572 ± 5e-5 |
| Higgs | 0.8468 | 0.8472 | 0.8470 | **0.8482 ± 3e-5** | 0.8475 ± 3e-5 |
| Epsilon | **0.9495** | 0.9465 | 0.9459 | 0.9469 ± 5e-5 | 0.9472 ± 5e-5 |
| MNIST8M | 0.8638 | 0.8636 | 0.8636 | 0.8649 ± 2e-5 | **0.8650 ± 2e-5** |

从准确性指标来看，在相同的迭代次数下，改进的算法在 4 个数据集上均优于基线算法，在全部数据集上的表现均优于 lgb_goss。其中 Flight Delay 和 Higgs 数据集上 lgb_ngoss 表现最优，但 lgb_enhanced 性能有所下降，合并箱体对准确性产生了影响，但由于分箱时对数据分布有了更多的感知，性能依然优于 lgb_goss。对 Epsilon 和 Higgs 数据集而言，二者都为稠密数据，但前者比后者更为平衡，算法在处理平衡数据时优势更为明显。同时，MNIST8M 数据集上算法的性能提升高过其它数据集，这表明算法在多分类问题上有较好的适用性。不同数据集的训练时间如表 4 所示。

**Table 4.** Comparison of training time across different algorithms
**表 4.** 不同算法的训练时间比较

| 数据集 | xgb_hist | lgb_baseline | lgb_goss | lgb_ngoss | lgb_enhanced |
|---|---|---|---|---|---|
| Allstate | 141.35 | 149.26 | 114.97 | 115.87 ± 0.52 | **114.10 ± 0.58** |
| Flight Delay | 63.01 | 62.15 | 51.73 | 52.02 ± 0.28 | **51.06 ± 0.35** |
| Higgs | 67.47 | 74.93 | 62.98 | 63.36 ± 0.47 | **60.16 ± 0.44** |
| Epsilon | 1183.70 | 323.61 | 148.09 | 150.07 ± 1.23 | **117.39 ± 1.02** |
| MNIST8M | 5256.68 | 6999.08 | 3090.01 | 3093.17 ± 2.12 | **2786.67 ± 2.33** |

lgb_enhanced 在所有数据集上均实现了训练时间的缩短。特别是在稠密数据和部分稀疏数据上速度提升明显。在 Epsilon 上，相比 lgb_goss，lgb_enhanced 训练时间从 148.09 秒减少到 117.39 秒，减少了 20.7%；在 MNIST8M 上，训练时间减少了 9.8%。

通过分析数据集特征与加速效果的关系，我们发现数据的稠密性和加速效果密切相关，稀疏数据的加速非常有限，最大箱体数设定为 255 的情况下，稀疏数据的大部分特征实现的分箱数本身就比 255 小得多，因而分箱合并对速度提升有限。对稠密数据和部分稀疏数据而言，大部分特征的总分箱数可以达到 255，因而使用分箱合并能显著减少遍历的总箱数，提升速度。同时，观察到 Higgs 数据集是稠密数据集，但它的速度只提升了 4.5%，原因在于它的特征数较少，进行分箱合并减少的总箱数有限。

观察表 3 与表 4 中 lgb_baseline、lgb_goss 和 lgb_enhanced 的结果，可以发现，依次加入采样改进和直方图改进，集成模型性能提升，训练时间缩短。其中，采样改进提升了所有数据集上集成模型的性能，直方图改进由于引入分箱合并产生了信息损失，只在 MNIST8M 数据集上有性能提升，但协同优化后的性能依然优于 lgb_baseline 和 lgb_goss。同时，采样改进在所有数据集上的训练速度提升都较为显著，而直方图改进在稠密数据和部分稀疏数据上效果显著，协同优化后的 LightGBM 取得了最优的训练效率。

**(2) NGOSS 方法分析**

观察表 3 和表 4 的结果，基于牛顿法的梯度单边采样引入二阶信息，用训练速度的微小下降提升了模型性能。为了进一步分析不同采样率下 NGOSS 的性能，在分类任务数据集 Higgs 和排序任务数据集 Istella LTR 上进行了进一步实验，实验使用 GOSS 和 Bagging 作为基线采样方法，在每一个采样率下使用不同的 top_rate (重要样本的保留比例)和 other_rate (剩余样本的采样比例)，Bagging 采样率为 top_rate 和 other_rate 之和。每一个采样率下，算法都经过了非常长的迭代并达到了收敛，同时实验采用了早停机制，Higgs 的早停轮数被设定为 50，Istella LTR 的早停轮数被设定为 60。图 2 是 Higgs 数据集不同采样率下的 AUC 值，图 3 是 Istella LTR 数据集不同采样率下的 NDCG@10 值。

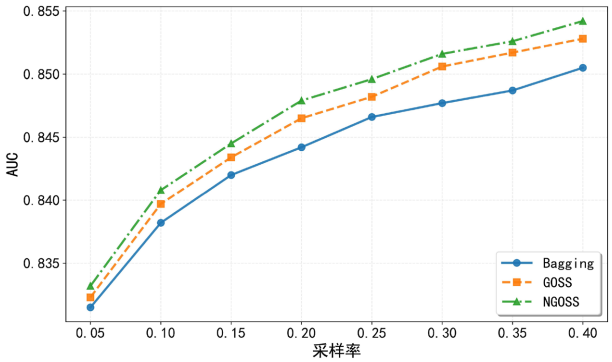


**Figure 2.** AUC values of algorithms under different sampling rates (using the Higgs dataset)
**图 2.** 不同采样率下算法的 AUC 值(使用 Higgs 数据集)

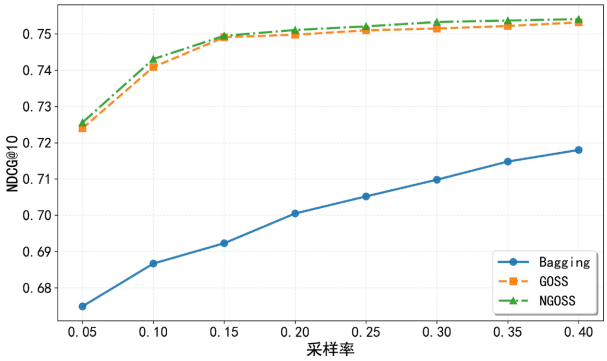


**Figure 3.** NDCG@10 values of algorithms under different sampling rates (using the Istella LTR dataset)
**图 3.** 不同采样率下算法的 NDCG@10 值(使用 Istella LTR 数据集)

可以观察到，每一个采样率下，本文提出的 NGOSS 方法都优于 GOSS 和 Bagging，这与 2.1 节的讨论相一致。对分类问题而言，三种采样方式的最终性能随采样率上升而上升，分类问题对三种采样方式的采样率都比较敏感；对排序问题而言，Bagging 方法在所有采样率下性能表现都较差，GOSS 和 NGOSS 方法在采样率小于 0.20 的情况下较为敏感，二者达到一定采样率会逐渐平稳，敏感性减弱，因而可以用较小的采样率(较少的计算量)达到较好的性能。所有的实验都表明 NGOSS 是比 GOSS 和 Bagging 精度更高的采样方式。图 4 和图 5 是 Higgs 数据集和 Istella LTR 数据集不同采样率下的训练时间，这里的训练时间是 600 次迭代的总训练时间。

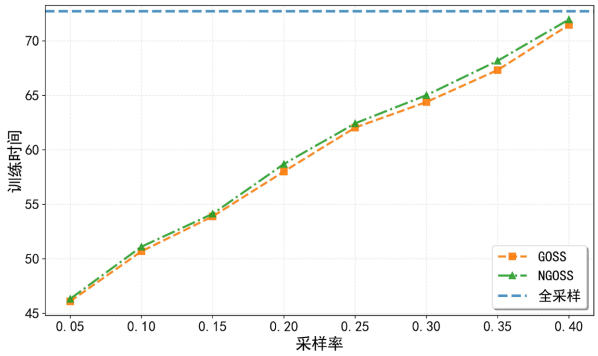


**Figure 4.** Training time of algorithms under different sampling rates (using the Higgs dataset)
**图 4.** 不同采样率下算法的训练时间(使用 Higgs 数据集)

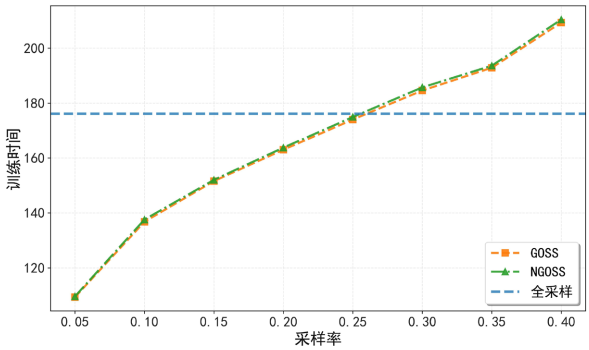


**Figure 5.** Training time of algorithms under different sampling rates (using the Istella LTR dataset)
**图 5.** 不同采样率下算法的训练时间(使用 Istella LTR 数据集)

可以观察到，训练时间随采样率的上升而上升，相比 GOSS，NGOSS 采样在引入了二阶信息后训练时间有所提升，但都维持在 1 秒左右。同时可以观察到，相比全采样，训练时间并没有随采样率的下降而线性下降，这是因为在进行 NGOSS 的过程中会涉及到全样本的计算，比如预测值的计算、一阶导数和二阶导数的计算等。另外，对不同的任务和不同的数据集，NGOSS 的 top_rate 和 other_rate 需要仔细选择，不合适的采样率可能导致训练时间超过全采样。

**(3) 直方图算法分析**

图 6 展示了 Epsilon 数据集上训练时间与 AUC 的关系曲线。

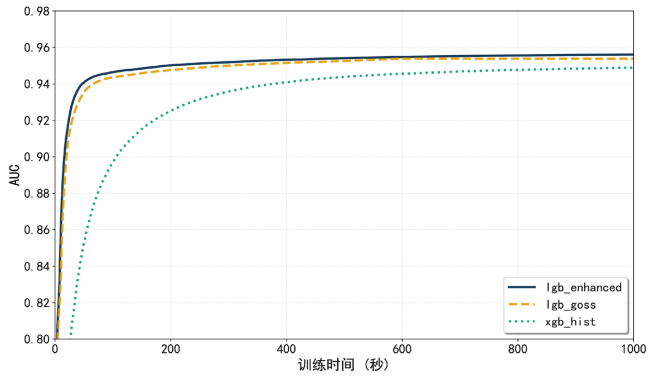


**Figure 6.** Training time-AUC curve for the Epsilon dataset
**图 6.** Epsilon 数据集的训练时间-AUC 曲线

lgb_enhanced 在整个训练过程中性能始终优于 lgb_goss 和 xgb_hist。特别在早期阶段(前 100 秒)，lgb_enhanced 的 AUC 上升速度明显快于 lgb_goss 和 xgb_hist，表明我们的改进方法具有更快的收敛速度。

定量分析显示，lgb_enhanced 的 AUC 达到最优性能的 95%所需的时间为 193.42 秒，而 lgb_goss 需要 298.54 秒，速度提升了 35.2%。比较达到 98%最优性能所需的时间，lgb_enhanced 比 lgb_goss 减少了 25.4%。这些结果表明我们的改进不仅提高了最终性能，更显著加快了模型收敛速度。

曲线形状分析发现，lgb_enhanced 的 AUC 曲线在早期阶段斜率更大，且更早达到平稳状态，这反映了优化后的直方图算法能更快找到有效的特征分割点，从而加速模型学习过程。图 7 展示了 MNIST8M 数据集上训练时间与准确率的关系曲线。

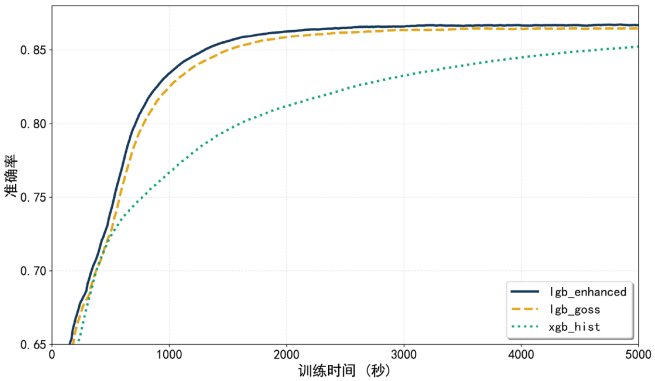


**Figure 7.** Training time-accuracy curve for the MNIST8M dataset
**图 7.** MNIST8M 数据集的训练时间－准确率曲线

与 Epsilon 数据集类似，lgb_enhanced 在 MNIST8M 上也展现出更快的收敛速度和更高的最终准确率。在训练初期(前 2000 秒)，lgb_enhanced 的准确率提升速度明显快于 lgb_goss，这对于实际应用中需要快速获得可用模型的场景尤为重要。定量比较显示，lgb_enhanced 达到 95%准确率所需时间比 lgb_goss 减少了 8.4%。

## 4. 结论与展望

本文提出了改进的数据处理方法，旨在提升 LightGBM 算法的预测准确性和计算效率。首先，我们设计了一种基于牛顿法的梯度单边采样技术，该方法充分利用二阶导数信息，提高了算法识别关键样本的能力。其次，我们提出了一种结合分布感知和标签感知的动态直方图算法，通过精准捕捉数据分布特征和标签信息，实现了性能提升和动态分箱，在保证高准确性的同时提升了计算速度。实验结果表明，改进后的算法在多种数据集上不仅实现了预测性能的提升，还在稠密数据集及部分稀疏数据集上显著提高了计算效率，充分验证了方法的有效性和实用价值。

本文提出的方法提升了 LightGBM 的性能和训练速度，但仍有一定的改进空间。针对牛顿采样增强，我们将进一步探索它的理论性质，同时与更多采样方法进行比较实验，证明其优越性。针对直方图分箱算法，本文提出的分布感知基于贪心策略，同时箱体合并只针对分类问题，如何实现全局分布感知和非分类问题箱体合并将作为下一步的研究切入点。

## 参考文献


[1] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W. and Liu, T. Y. (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*, **30**, 3147-3155.

[2] Friedman, J.H. (2001) Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, **29**, 1189-1232. https://doi.org/10.1214/aos/1013203451

[3] Ponsam, J.G., Bella Gracia, S.V.J., Geetha, G., Karpaselvi, S. and Nimala, K. (2021) Credit Risk Analysis Using LightGBM and a Comparative Study of Popular Algorithms. 2021 4*th International Conference on Computing and Communications Technologies* (*ICCCT*), Chennai, 16-17 December 2021, 634-641. https://doi.org/10.1109/icoct53315.2021.9711896

[4] Ge, D., Gu, J., Chang, S. and Cai, J. (2020) Credit Card Fraud Detection Using LightGBM Model. 2020 *International Conference on E-Commerce and Internet Technology* (*ECIT*), Zhangjiajie, 22-24 April 2020, 232-236. https://doi.org/10.1109/ecit50008.2020.00060

[5] Han, L., Yang, T., Pu, X., Sun, L., Yu, B. and Xi, J. (2021) Alzheimer's Disease Classification Using LightGBM and Euclidean Distance Map. 2021 *IEEE* 5*th Advanced Information Technology*, *Electronic and Automation Control Conference* (*IAEAC*), Chongqing, 12-14 March 2021, 1540-1544. https://doi.org/10.1109/iaeac50856.2021.9391046

[6] Alzamzami, F., Hoda, M. and El Saddik, A. (2020) Light Gradient Boosting Machine for General Sentiment Classification on Short Texts: A Comparative Evaluation. *IEEE Access*, **8**, 101840-101858. https://doi.org/10.1109/access.2020.2997330

[7] Ong, Y.J., Zhou, Y., Baracaldo, N. and Ludwig, H. (2020) Adaptive Histogram-Based Gradient Boosted Trees for Federated Learning.

[8] Zhang, H., Si, S. and Hsieh, C.J. (2017) GPU-Acceleration for Large-Scale Tree Boosting.

[9] Meng, Q., Ke, G., Wang, T., Chen, W., Ye, Q., Ma, Z.M. and Liu, T.Y. (2016) A Communication-Efficient Parallel Algorithm for Decision Tree. *Advances in Neural Information Processing Systems*, **29**, 1279-1287.

[10] Shi, Y., Ke, G., Chen, Z., Zheng, S. and Liu, T. Y. (2022) Quantized Training of Gradient Boosting Decision Trees. *Advances in Neural Information Processing Systems*, **35**, 18822-18833.

[11] Chen, T. and Guestrin, C. (2016) XGBoost. *Proceedings of the* 22*nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, 13-17 August 2017, 785-794. https://doi.org/10.1145/2939672.2939785