Operations Research and Fuzziology 运筹与模糊学, 2024, 14(6), 266-279
Published Online December 2024 in Hans. https://www.hanspub.org/journal/orf
https://doi.org/10.12677/orf.2024.146529

Hans汉斯

基于改进梯度下降算法的XGBoost模型研究

黄　丽

南京信息工程大学数学与统计学院，江苏　南京

收稿日期：2024年9月17日；录用日期：2024年11月27日；发布日期：2024年12月6日

摘　要

分数阶微积分的优良性质为基于梯度下降的优化方法提供了新的前景，相比于传统整数阶，分数阶微积分可以通过不断更新来调整阶数。XGBoost模型结合了正则化技术和梯度提升决策树，具有高效性和准确性。但在实际应用中，XGBoost模型仍存在训练速度较慢、易过拟合等局限性。为了解决这些问题，本文所做工作主要有如下几个方面：1) 基于Caputo定义引入了分数阶微积分的基本内容。作为一种经典的凸优化算法，梯度下降法在结合分数阶微积分理论后延伸出了分数阶梯度下降算法。2) 在简化目标函数的过程中，将分数阶梯度下降算法与XGBoost模型相结合，并选择合适的分数阶，得到新的分数阶XGBoost模型，在保证准确性的同时提高了训练速度。3) 利用改进后的XGBoost模型对房价预测数据集进行实证分析。与传统的整数阶XGBoost模型进行对比分析，改进后的模型具有更快的收敛速度和更好的预测性能。

关键词

XGBoost，随机梯度下降，房价预测，分数阶微积分

Research on XGBoost Model Based on Improved Gradient Descent Algorithm

Li Huang

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

Received: Sep. 17th, 2024; accepted: Nov. 27th, 2024; published: Dec. 6th, 2024

Abstract

The excellent properties of fractional calculus provide a new prospect for optimization methods based on gradient descent, which can be continuously updated to adjust the order compared to the traditional integer order. The XGBoost model combines regularization techniques and gradient boosting decision trees, which is efficient and accurate. However, in practical applications, the

文章引用: 黄丽. 基于改进梯度下降算法的 XGBoost 模型研究[J]. 运筹与模糊学, 2024, 14(6): 266-279.
DOI: 10.12677/orf.2024.146529

XGBoost model still has limitations such as slow training speed and easy overfitting. In order to solve these problems, the work done in this paper is mainly as follows: 1) Based on the Caputo definition, the basic content of fractional calculus is introduced. As a classical convex optimization algorithm, the gradient descent method is extended to the fractional gradient descent algorithm after combining the fractional calculus theory. 2) In the process of simplifying the objective function, the fractional gradient descent algorithm is combined with the XGBoost model, and the appropriate fractional order is selected to obtain a new fractional XGBoost model, which improves the training speed while ensuring the accuracy. 3) The improved XGBoost model is used to conduct an empirical analysis of the housing price prediction dataset. Compared with the traditional integer order XGBoost model, the improved model has faster convergence speed and better prediction performance.

Keywords

XGBoost, Stochastic Gradient Descent, Housing Price Forecast, Fractional Calculus

1. 引言

1.1. 研究的现状及发展趋势

梯度法自诞生以来就备受瞩目并获得飞速发展，机器学习领域里，梯度下降法是迄今为止解决无约束凸优化问题最简单有效的方法，当前神经网络训练的优化算法也主要与梯度下降算法有关。该算法有着简单通用的特点，然而随着数据规模日渐增大，传统梯度下降法的收敛特性有时并不能取得令人满意的结果，为了进一步改善梯度下降法的收敛性，许多改进梯度下降算法已经出现[1]。

梯度下降法在 1874 年首次被数学家柯西提出，该算法常用于求解无约束凸优化问题，它有着容易实现、构造方便、直观等优势[2]。虽然在优化理论中梯度下降法是一个经典的方法，但是传统的梯度法存在“之”字形低效率搜索的缺陷以及在极值周围收敛速度较慢的劣势。为了处理梯度下降方法的不足，研究人员开发了各种梯度下降优化算法。分数阶梯度下降算法是一种新型的优化算法，相比于传统整数阶梯度下降算法，具有更大的自由度，分数阶微积分可以通过更新来不断地调整阶数，从而为提高梯度下降算法的性能提供新的可能[3]。

文献[4]提出了牛顿迭代结合梯度下降算法的方法，以此来提高过程神经网络的训练效率。文献[5]提出了一种变阶次分数阶梯度下降算法，能够确保在不改变微分下限的条件下，真实的目标函数仍能收敛到极值点。文献[6]提出了一种分数阶自适应学习方法，然而结果显示，虽然算法最终是收敛的，却不能保证找到最优解。文献[7]提出的自适应最小均方算法正好克服了这一缺陷，确保了这种方法是有效的信号处理方式。在 Riemann-Liouville 定义的分数阶导数基础上，文献[8]提出了更加具有实际使用价值的变初值分数阶梯度下降算法，以确保能收敛到实际的极值，引入了随机权重粒子群算法来选取合适的初始值，这种改进有效地克服了局部最优、精度和收敛速度等问题。文献[9]陈述了可用于神经网络 BP 训练的分数阶梯度下降方法，利用 Caputo 导数估计误差分数梯度。文献[10]基于分数阶梯度下降算法和改进的 Riemann-Liouville 导数的凸组合，提出了一种新的分数梯度下降学习算法(FOGD)的径向基函数神经网络。文献[11]提出了一种自适应分数阶 BP 神经网络，将优化后的竞争进化算法与分数阶梯度下降学习机制相结合，用以解决手写体数字识别问题。文献[12]针对局部最优的问题，提出采用分数阶动量的思想，

将基于动量的随机梯度下降法与分数阶差分运算相结合，改进了参数更新方法，并给出阶次调整方法，提出了基于分数阶动量的随机梯度下降法，实验结果表明，这种方法确实能够改善卷积神经网络的精度和收敛速度。分数阶梯度在非线性科学中的应用也是一个研究热点，近年来，分数阶梯度法已被应用到神经网络、机器学习、控制系统、图像处理等领域。

极端梯度提升决策树(Extreme Gradient Boosting)简称 XGBoost，是一种基于残差优化的大规模并行提升树的工具，在处理结构化数据的任务中被广泛使用，它在目标函数中加入正则化并且采取列抽样，既避免了模型过度拟合，又显著提高了运算速度。XGBoost 模型在收敛速度快的情况下不会损失模型的鲁棒性，因此被认为是目前最好的机器学习算法之一。然而，XGBoost 模型在某些情况下可能存在一些缺陷。例如，当训练集较大或特征数量较多时，训练速度会变得非常缓慢。此外，XGBoost 在过拟合方面也存在一些问题。

1.2. 研究的意义和价值

梯度下降法研究的重点是用它来确定学习率和选择优化方向。近年来为了降低计算复杂度，随机梯度下降算法已经成为机器学习特别是深度学习的研究热点。历经近几年的有关探索，国内外取得了巨大的研究成果[13]。

XGBoost 是一种被广泛使用的机器学习算法，尤其适用于大规模数据集和复杂任务。此算法的基本思想是首先建立一个基分类器，然后逐渐增加一个新的分类器，并计算添加新分类器后其目标函数的值，从而保证在迭代过程中，目标函数值逐渐减小，不断提高模型的表达效果[14]。在传统的 XGBoost 模型中，梯度下降算法是用于优化损失函数的重要方法，然而在某些情况下，梯度下降算法可能会遇到局部最优解或者收敛速度较慢的问题。

分数阶梯度下降算法是一种新型的优化算法，可以有效地克服 XGBoost 模型中所存在的问题，该算法不仅可以适应各种损失函数，而且可以通过调整分数阶导数的值来平衡全局搜索和局部搜索。因此，我们提出了一种基于分数阶梯度下降算法的 XGBoost 模型，并将其应用于某些任务中。

2. 预备知识

2.1. 梯度下降法

梯度下降法的主要思想是通过逐渐增加迭代次数来调整损失函数的权重，使损失函数最小化。传统的整数阶梯度下降方法是沿目标函数的负梯度方向迭代更新，最终收敛到极小值点的过程。对于无约束凸优化问题：

$$\min_{x\in R^n} f(x), \tag{1}$$

可微凸函数 $f(x)$ 在全局有唯一的极小值点 x^*， $f(x^*)$ 为当 $x=x^*$ 时的极小值。对于这类问题，梯度下降算法能够通过迭代迅速寻找到极小值点，迭代公式为：

$$x_{k+1}=x_k-\eta\nabla f(x_k), \tag{2}$$

迭代的步长 $\eta>0$，$\nabla f(x)$ 是 $f(x)$ 在 x 处的梯度，迭代步数为 k，第 k 步的变量值为 x_k。需要注意的是，步长 η 若选择的过大可能会使算法不收敛，选择的步长过小又会使算法的收敛速度变得非常慢。此外，若目标函数中有多个极值点，用梯度下降法计算的结果很可能只得到一个局部极值点。

2.2. 分数阶微积分

分数阶导数有 3 种被广泛接受的定义，例如 Riemann-Liouville、Grünwald-Letnikov、Caputo，Caputo

分数阶导数是一种常用的非整数阶导数，其定义如下[15]：

$$ {}_{c}^{C}D_{x}^{\alpha} f(x) = \frac{1}{\Gamma(n-\alpha)} \int_{c}^{x} \frac{f^{(n)}(\tau)}{(x-\tau)^{\alpha-n+1}} \mathrm{d}\tau, \tag{3} $$

其中 $n-1<\alpha<n$， $n\in\mathbb{Z}_{+}$，积分下限是 c，积分区间是 $[c,x]$，Gamma 函数为 $\Gamma(\alpha)=\int_{0}^{+\infty} \mathrm{e}^{-t}t^{\alpha-1}\mathrm{d}t$。

特别地，当 $n=1$，即 $\alpha\in(0,1)$ 时，有：

$$ {}_{c}^{C}D_{x}^{\alpha} f(x) = \frac{1}{\Gamma(1-\alpha)} \int_{c}^{x} \frac{f'(\tau)}{(x-\tau)^{\alpha}} \mathrm{d}\tau. \tag{4} $$

如果足够光滑且其分数阶微分存在有限，对其 Caputo 分数阶微分做无穷次分部积分运算，有：

$$ \begin{aligned} {}_{c}^{C}D_{x}^{\alpha} f(x) &= \sum_{i=1}^{+\infty} C_{\alpha-1}^{i-1} \frac{f^{(i)}(x)}{\Gamma(i+1-\alpha)} (x-c)^{i-\alpha} \\ &= \sum_{i=1}^{+\infty} \frac{f^{(i)}(c)}{\Gamma(i+1-\alpha)} (x-c)^{i-\alpha}, \end{aligned} \tag{5} $$

一般而言，分数阶导数可以看作是传统导数的推广。

2.3. 决策树原理

决策树(Decision tree)是一种常见的基学习器，作为一种树状预测模型，其结构见图 1。

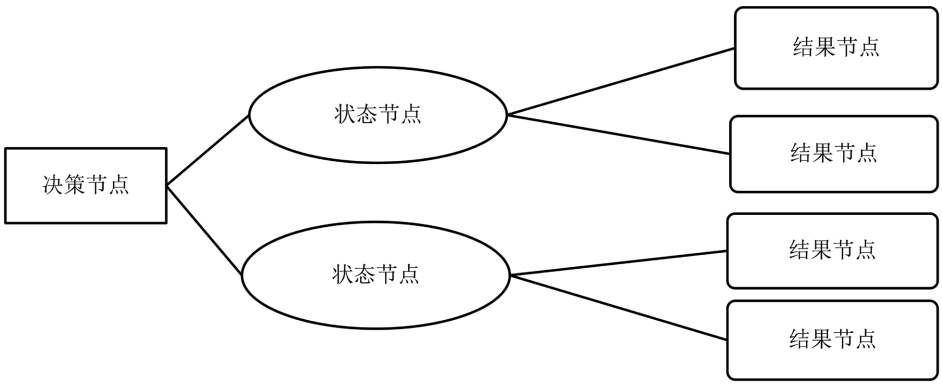

Figure 1. Diagram of the basic structure of a decision tree
图 1. 决策树基本结构图

在机器学习有关范围内，决策树是一种预测模型，每个内部节点代表对属性的一个判断，根据判断是否满足形成二叉树，然后根据其他判断继续向下输出，最终每个叶子节点代表一个分类结果[16]。测试属性的选择以及样本集的划分是构成决策树的关键，分类树算法的根本区别是属性选取标准的不同。

决策树生成算法递归生成决策树，直到无法终止。通过这种方式生成的决策树一般来说对训练数据集分类精确，然而对于未知数据集分类精度不高，存在严重的过拟合状况。所以，为了降低模型的复杂性，使模型的泛化能力更强，需要对生成的决策树进行剪枝。决策树模型在分类问题中有许多优点，例如可以用图形表示、处理数据简单、易于理解和实现等。同时，决策树还能处理具有不相关特征的数据，而且能够方便地构造易于理解的规则。因此，XGBoost 已成为数据科学竞赛中最受欢迎的算法之一，也被广泛应用于实际工业界的数据挖掘和预测任务中。决策树模型也存在一些缺陷，例如对连续性的字段比较难预测、过拟合和忽略数据集中属性之间的相关性。本论文将会使用 XGBoost 模型来进行研究，将

XGBoost 模型与改进梯度下降算法相结合，得到基于改进梯度下降算法的 XGBoost 模型，并与传统的 XGBoost 模型进行对比。

3. XGBoost 模型及其改进

3.1. 分数阶梯度下降算法

基于传统的梯度下降法，改进梯度下降算法是通过引进一些优化方法，以此来提升算法的收敛速度和性能。分数阶微分是一个非局部算子，所以可能导致分数阶梯度法不能收敛到极值点，因此必须要消除分数阶微分的长记忆性。基于 Caputo 定义，本文将使用一种迭代初始值策略设计的分数阶梯度下降算法，从而在算法收敛时消除分数阶微分的非局部性质，使其收敛到真正的极值点[17]。

众所周知梯度下降法是一种用于查找函数最小值的一阶迭代优化算法。为此，人们通常会采取与当前点函数梯度(或近似梯度)的负数成正比的步骤。迭代初始值的 Caputo 定义下的分数阶梯度下降算法如下[17]：

$$x_{k+2} = x_{k+1} - \mu {}_{x_k}^{C}D_{x_{k+1}}^{\alpha} f(x), 0 < \alpha < 1, \tag{6}$$

(6)式中的分数阶导数其实是一种特殊的积分，体现了函数 $f(x)$ 的长记忆性。

分数阶梯度下降算法中 x_k 根据以下公式更新：

$$x_{k+1} = x_k - \mu \nabla f(x_k), \tag{7}$$

其中，x_k 为第 k 步的变量值，μ 是学习率 $\nabla f(x_k)$ 是 $x = x_k$ 时的一阶梯度，即 $\frac{\mathrm{d}}{\mathrm{d}x} f(x)\Big|_{x=x_k}$，分数阶梯度下降算法的流程如表 1 所示。当一阶梯度被分数梯度取代时，x_k 服从：

$$x_{k+1} = x_k - \mu \nabla^{\alpha} f(x_k). \tag{8}$$

Table 1. Fractional gradient descent algorithm
表 1. 分数阶梯度下降算法

算法：分数阶梯度下降算法(FOGD)
1：初始化 $\alpha, n = k, n_{\max}, \mu, \gamma$，以及 $x_0, x_1, \cdots, x_k$；
2：计算在 x_n 处的 α 阶梯度
$${}_{x_{n-k}}^{C}D_{x_n}^{\alpha} f(x_n) = \frac{2a^2}{\Gamma(3-\alpha)}(x_n - x_{n-k})^{2-\alpha} + \frac{2a(ax_{n-k}+b)}{\Gamma(2-\alpha)}(x_n - x_{n-k})^{1-\alpha}$$
3：按照 $x_{n+1} = x_n - \mu {}_{x_{n-k}}^{C}D_{x_n}^{\alpha} f(x_n)$ 更新参数
4：将 n 增加 k，然后检查 $f(x_n)$ 是否满足精度要求，或者当 $n > n_{\max}$ 时，退出算法；若是不满足，则继续步骤 2。

3.2. XGBoost 模型

XGBoost 是损失函数的二阶泰勒展开式，除了一般的损失函数，还加入一个正则项来控制模型的复杂性。它可以计算整体的最优解，衡量损失函数的下降和模型的复杂性，避免过拟合，从而提高模型的效率[18]。XGBoost 是一个提升树模型，它将许多树模型聚合在一起形成一个强分类器。整体结构是基于一棵树的训练和下一棵树的训练来预测其与真实分布的差距，通过不断训练来弥补差距的树，最后组合树来实现对真实分布的模拟。算法的基本原理如下：

设$(x_i, y_i), i=1,2,\cdots,n$是建模样本，$\hat{y}_i^{(K)}$是第 K 轮的预测值，等于前$K-1$轮预测值加上新的函数$f_K(x_i)$，$f_K(x_i)$是第 K 个决策树的预测结果，x_i是第 i 个输入数据。作为一种提升树，XGBoost 模型的假设空间也是一组分类回归树(CART)的集成，输出为

$$x_{k+1} = x_k - \mu\nabla^{\alpha} f(x_k). \tag{9}$$

其模型参数为 K 棵树，

$$\mathbb{F} = \{f_1, f_2, \cdots, f_K\}, \tag{10}$$

$$\hat{y}_i^{(K)} = \hat{y}_i^{(K-1)} + f_j(x_i). \tag{10}$$

XGBoost 的目标函数包括两部分：损失函数部分 L 和正则化项部分 Ω。损失函数是用来衡量模型在训练数据上的适合程度，正则化项是用来衡量模型的复杂度。在第 K 轮迭代时 $K-1$ 轮的预测结果是固定的，因此模型目标函数的设定只需要考虑预测函数。对于 XGBoost 模型，若数据集 D 中有 n 个样本，每个样本有 m 维特征，通过训练数据集 D，我们可以得到 K 棵树。这 K 棵树累加的值就是预测值 $\hat{y}_i$。XGBoost 的目标函数为[14]。

$$Obj(x_i) = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{j=1}^{K} \Omega(f_j), \tag{12}$$

其中，$L(y_i, \hat{y}_i)$是损失函数，根据具体的问题，损失函数可以做不同的设定，例如，回归是 MSE，分类是交叉熵。$\Omega(f_j)$是树的复杂度，也是正则项，通过统计所有树模型复杂度的相关参数，可减少过拟合。若$\Omega(f)=0$，则 XGBoost 模型退化为具有 2 个变量的 Boosting 模型。

由于 XGBoost 是一个加法模型，因此，预测值是每棵树预测值的累加之和，这点与传统提升树一样。在 K 次迭代中，可以将树展开成：

$$\hat{y}_i^{(K)} = \sum_{j=1}^{K} f_j(x_i) = f_1(x_i) + f_2(x_i) + \cdots + f_{K-1}(x_i) + f_K(x_i), \tag{13}$$

但训练第 K 棵树的时候，目标函数 Obj 可以表示为：

$$Obj^K(x_i) = L + \Omega = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(K-1)} + f_K(x_i)\right) + \sum_{j=1}^{K-1} \Omega(f_t) + \Omega(f_K). \tag{14}$$

CART 决策树是 XGBoost 中的基学习器，每个叶子节点对应的都有一个预测值，也称之为叶子节点的得分或者权重值，对于每棵 CART 树，每一棵树的结构为 q，它可以将样本映射到对应的叶节点索引上；$\omega_j\ (1 \le j \le K)$为每个叶子节点的权重值，所以，某个 CART 树可以表示为：

$$f_i = \left\{\omega_q(x_i) \mid q : R^m \to T, \omega \in R^T\right\}, \tag{15}$$

去除已知项，则目标函数 Obj 变为：

$$Obj^K(x_i) = \sum_{i=1}^{n} L\left[y_i, \hat{y}_i^{(K-1)} + f_K(x_i)\right] + \Omega(f_K). \tag{16}$$

若叶子节点数过多，模型容易陷入过拟合，因此需要指定惩罚项来限制叶子节点的个数，惩罚项表达式如式(17)所示：

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2. \tag{17}$$

决策树叶子节点个数为 T，它是对应叶子节点的预测值(叶子节点的平方再求和)，γ为惩罚强度，λ

是对应的调整系数。将损失函数进行泰勒展开到二次项，使用贪婪算法或者其它算法可以求解模型的参数。损失函数 $L\left(y_i,\hat{y}_i\right)$ 与传统提升树的损失函数一样，可以在回归中使用平方误差，在分类中使用对数损失。

3.3. 基于分数阶梯度下降算法的 XGBoost 模型

XGBoost 有自己的独特性。训练模型一般是先定义一个目标函数，然后对其进行优化。XGBoost 的目标函数包括损失函数和正则项两个方面，$F=L+\Omega$。损失函数表示模型数据的拟合程度，我们通常用它的一阶导数来指出梯度下降的方向，XGBoost 除此之外又计算了它的二阶导数，而且接着考虑梯度变化的趋势。正则项用于控制模型的复杂性，叶节点数量越多，模型就越大，这样的话不仅运算时间变长，而且超过一定限度后，会出现过拟合，导致分类效果下降。XGBoost 的正则项是一种惩罚机制，叶子节点越多，惩罚越大，从而限制了它们的数量。XGBoost 在数学原理之外最大的进步就是大大提高了计算速度。擅长发现复杂数据之间的依赖关系，可以从大规模数据集中获得有效的模型，在实际应用中支持多种系统和语言。

原本的 XGBoost 模型是二阶优化算法，也就是说求解目标函数的二阶导数，它具有一定的局限性。相比于经典梯度下降法，分数阶梯度下降算法收敛到极值点的速度更快，所以可以利用分数阶梯度下降算法所计算出的数值来诱导产生新的决策树。用分数阶梯度下降算法优化后的模型可以和原本的 XGBoost 模型解决同样的问题并且优化后的模型收敛速度更快。

一般的二阶泰勒展开式为：

$$f\left(x+\Delta x\right)\approx f\left(x\right)+f'\left(x\right)\Delta x+\frac{1}{2}f''\left(x\right)\Delta x^2, \tag{18}$$

对式(16)依变量，进行二阶泰勒展开得到近似目标函数：

$$\begin{aligned} Obj^K\left(x_i\right)&\approx\sum_{i=1}^{n}\left[L\left(y_i,\hat{y}_i^{(K-1)}\right)+\partial_{\hat{y}^{(K-1)}}L\left(y_i,\hat{y}_i^{(K-1)}\right)f_K\left(x_i\right)+\frac{1}{2}\partial^2_{\hat{y}^{(K-1)}}L\left(y_i,\hat{y}_i^{(K-1)}\right)f_K^2\left(x_i\right)\right]+\Omega\left(f_K\right)\\ &=\sum_{i=1}^{n}\left[L\left(y_i,\hat{y}_i^{(K-1)}\right)+g_i f_K\left(x_i\right)+\frac{1}{2}h_i f_K^2\left(x_i\right)\right]+\Omega\left(f_K\right), \end{aligned} \tag{19}$$

其中 $g_i=\partial_{\hat{y}^{(K-1)}}L\left(y_i,\hat{y}_i^{(K-1)}\right)$，$h_i=\partial^2_{\hat{y}^{(K-1)}}L\left(y_i,\hat{y}_i^{(K-1)}\right)$。

有必要说明的是 $L\left(y_i,\hat{y}_i^{(K-1)}\right)$ 是一个常数，在最小化时可以忽略，则优化后的目标函数简化为：

$$\tilde{Obj}^K\left(x_i\right)=\sum_{i=1}^{n}\left[g_i f_K\left(x_i\right)+\frac{1}{2}h_i f_K^2\left(x_i\right)\right]+\Omega\left(f_K\right), \tag{20}$$

由整数阶变为分数阶是将 $g_i=\dfrac{\partial L\left(y_i,\hat{y}_i^{(K-1)}\right)}{\partial\hat{y}_i^{(K-1)}}$ 替换为 $g_i=\dfrac{\partial L\left(y_i,\hat{y}_i^{(K-1)}\right)}{\partial\hat{y}_i^{(K-1)}}\dfrac{\left[\left|\hat{y}_i^{(K-1)}-c\right|+\varepsilon\right]^{1-\alpha}}{\Gamma\left(2-\alpha\right)}$。

定义 $I_j=\left\{i\,|\,q\left(x_i\right)=j\right\}$ 是叶节点 j 上的一组实例，由于预测结果是叶节点的输出，其中每个叶节点都会有一个权重，即输出 ω_j，所以我们可以改写公式(20)，化简后的目标函数是：

$$\begin{aligned} \tilde{Obj}^K\left(x_i\right)&=\sum_{i=1}^{n}\left[g_i f_K\left(x_i\right)+\frac{1}{2}h_i f_K^2\left(x_i\right)\right]+\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}\omega_j^2\\ &=\sum_{j=1}^{T}\left[\left(\sum_{i\in I_j}g_i\right)\omega_i+\frac{1}{2}\left(\sum_{i\in I_j}h_i+\lambda\right)\omega_i^2\right]+\gamma T, \end{aligned} \tag{21}$$

因为 $f_i=\left\{\omega_q\left(x_i\right)\,|\,q:R^m\to T,\omega\in R^T\right\}$，所以 $f_K\left(x_i\right)=\left\{\omega_j\big|_{q(x_i)=j}\right\}$，替换 f_K 得到式(21)。

对于固定的树结构 $q(x)$，我们可以进行求导，然后取零来推导出叶节点 j 的最优权重 ω_j，令 $G_j=\sum_{i\in I_j} g_i$，$H_j=\sum_{i\in I_j} h_i$，则

$$\tilde{Obj}^K(x_i)=\sum_{j=1}^{T}\left[G_j\omega_i+\frac{1}{2}\left(H_j+\lambda\right)\omega_i^2\right]+\gamma T, \tag{22}$$

$$F_{\omega_i}^{*}=G_j\omega_i+\frac{1}{2}\left(H_j+\lambda\right)\omega_i^2, \tag{23}$$

对 ω_j 进行求导，得到 $G_j+\left(H_j+\lambda\right)\omega_j=0$，因此 $\omega_j=-\frac{G_j}{H_j+\lambda}$，计算得出这棵决策树的目标函数最优值：

$$\tilde{Obj}^K(x_i)=-\frac{1}{2}\sum_{j=1}^{T}\left[\frac{G_j^2}{H_j+\lambda}\right]+\gamma T, \tag{24}$$

这样就可以得到叶子节点的权重值，从而可以求得目标函数，显然目标函数越小越好。这个公式可以看作是测量树结构 q 质量的衡量函数，类似使用基尼系数来衡量一棵树的质量。

4. 实例分析

4.1. 预测实例

4.1.1. 数据预处理

为了验证新模型的效果，本文将来自 Kaggle 比赛所使用的数据集应用于基于改进梯度下降算法的 XGBoost 模型中。Kaggle 比赛的题目是基于 Ames, Iowa 的房屋历史成交数据来预测新的房屋销售价格。

采用 XGBoost 模型解决基于 Ames, Iowa 的房屋历史成交数据预测房屋销售价格的问题，有以下几个理由：

1) 高效性：XGBoost 在执行速度上非常快，能够有效处理大规模的数据集。这使得它特别适合用于 Kaggle 比赛中常见的复杂问题。

2) 正则化功能：XGBoost 内置了 L1 和 L2 正则化，有助于减少过拟合，提高模型的泛化能力，尤其在处理高维数据时表现出色。

3) 支持缺失值处理：XGBoost 能够自动处理缺失值，这对于房屋销售数据来说是一个重要特性，因为数据集中可能存在缺失的属性。

4) 灵活性：XGBoost 支持多种损失函数，可以根据具体问题选择最适合的损失函数，从而提升模型性能。

首先对数据集进行处理，划分为训练集和测试集，训练集与测试集之间的区别在于只有训练集包括房子的价格，即标签。测试集包括每栋房子的特征，例如建造年份、街道类型、房价类型。不同特征对房屋价格的影响程度也有所不同，对于地皮面积来说，面积越大则房价也就相应的越高；而对于市区物理位置而言，区位越高，房价越高；除此之外还有整体材料和饰面质量，需要对房子的整体材料和装修进行评估计算。特征值有离散标签、连续数、缺失值(NA)等。

在进行数据处理中，之所以需要排除 y(房屋价格)，是因为该列在测试集中不存在。在测试集中预测 y 是我们的工作。由于原本数据集行列数过多，不宜加载，本文对其进行删除部分内容操作，以便于测试模型拟合程度。对于一些缺失值，常见的缺失值补全方法有均值插补、建模预测、同类均值插补、和极大似然估计等，本文采用均值插补、同类均值插补两种方法来填补缺失值。

4.1.2. 相关性分析

对于处理后的数据进行分析，研究各个变量与房屋真实价格之间的相关性见图 2，由于因素过多，在计算出相关性后选择出具有代表性的几个因素进行对比。

Table 2. Factor correlation analysis
表 2. 因素相关性分析

数据	OverallQual	GrLivArea	GargeCars	TotalBathrooms	TotalFlrSF	TotalBsmtSF
相关性	0.796	0.704	0.641	0.637	0.597	0.640

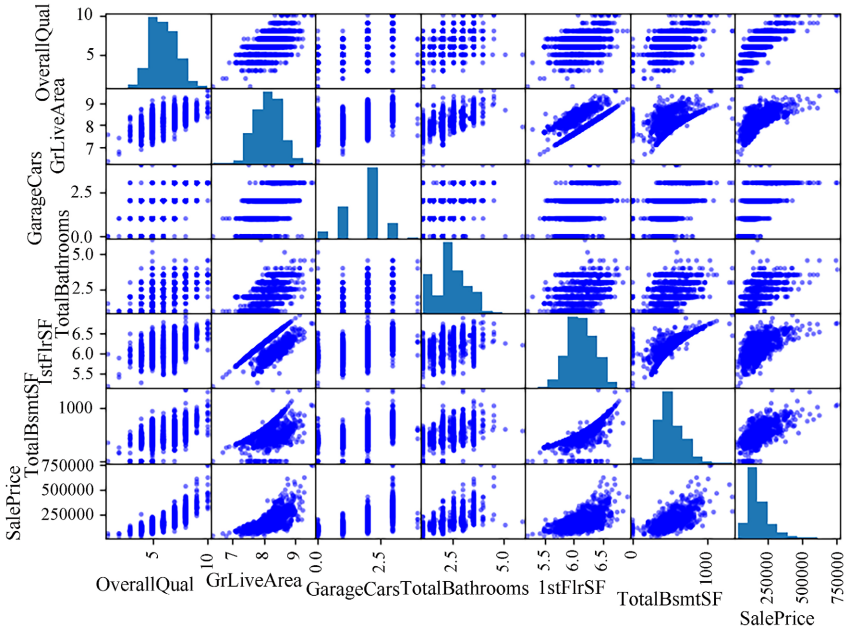

Figure 2. Factor correlation analysis
图 2. 因素相关性分析

一般 0.7 以上说明关系非常紧密；0.4~0.7 之间说明关系紧密；0.2~0.4 说明关系一般。初步探索一下这些因素之间是否存在关联，由表 2 可以看出，房屋价格与材料(Overall Qual)、房屋占地面积(GrLiv Area)与房屋价格之间有着非常显著的正相关关系，车库汽车(Garage Cars)、车库面积(Garage Area)、浴室面积(TotalBathrooms)等也与房屋价格之间有着正相关关系。

4.1.3. 建模过程

本文所采用的软件是 Python 语言，以此来实现 XGBoost 模型。模型预测步骤如下：

1) 将包含 1457 个数据的样本集进行数据预处理；

2) 分别构造未改进的 XGBoost 模型、改进后未调参的 XGBoost 模型；

3) 对改进后未调参的 XGBoost 模型进行调整参数；

对应的预测流程图如图 3 所示。

XGBoost 模型的建模包括 3 种参数：常规参数、基础模型参数和学习任务参数。研究的目标最终可以归结为回归问题，基础模型参数是对模型效果影响较大的部分，也是参数调整的关键，本文依据参数在模型中的重要程度对参数进行了调整。为了避免过拟合与欠拟合问题的出现，在训练集上以识别准确率为评价指标，针对 learning_rate 与 n_estimators 进行调参，其中，n_estimators 基于实际情况设置为 170，

max_depth 设置为 3，alpha 设置为 0.9，其他选项缺省。调参结果如表 3 所示。

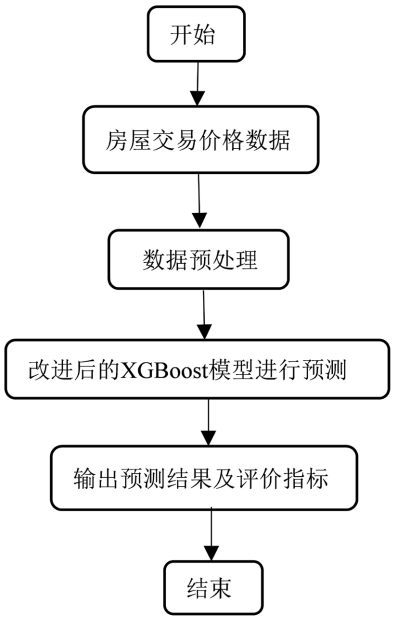

Figure 3. Predictive flowchart
图 3. 预测流程图

Table 3. Parameter adjustments
表 3. 参数调整

参数	含义	最优值
Learning_rate	学习率	0.1
n_estimators	回归器的数量	170
max_depth	树的最大深度	3
gamma	叶节点进一步划分所需的最小损失减少量	0.5
alpha	L1 正则化项	0.9
min_child_weight	生成一个子节点所需要的最少损失减少量	1
colsample_bytree	建立树时对特征随机采样的比例	1

4.2. 不同模型预测结果与分析

可以根据评价指标的高低来判断模型的质量，回归模型中没有最优模型这一说法，但可以通过对比评价指标从现有的模型中选择较好的模型。回归预测问题中常用的评价指标有平均绝对误差(MAE)、均方误差(MSE)、平均绝对百分比误差(MAPE)和拟合优度(R^2)等，这四个评价指标计算公式分别如下：

$$\mathrm{MAE}=\frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i-y_i\right| \tag{25}$$

$$\mathrm{MSE}=\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i-y_i\right)^2 \tag{26}$$

$$\mathrm{MAPE}\left(y_i,\hat{y}_i\right)=\frac{1}{n}\sum_{i=1}^{n}\frac{\left|y_i-\hat{y}_i\right|}{y_i}\times 100\% \tag{27}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(\bar{y} - y_i)^2} \tag{28}$$

其中，预测值为 $\hat{y}_i$，真实值为 y_i，真实值的期望为 $\bar{y}$。

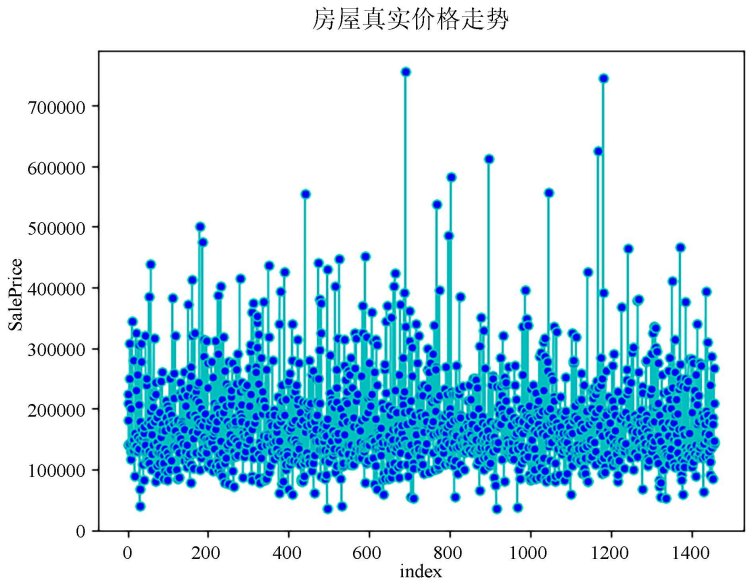

Figure 4. Real house price chart
图 4. 房屋真实价格走势图

数据预处理后的房屋实际价格走势见图 4，横坐标为房屋，纵坐标为房屋销售价格。导出各个模型得到的预测值，将其与真实值进行对比，可以侧面帮助我们分析改进前后模型的优劣，真实值与预测结果的对比如下所示。

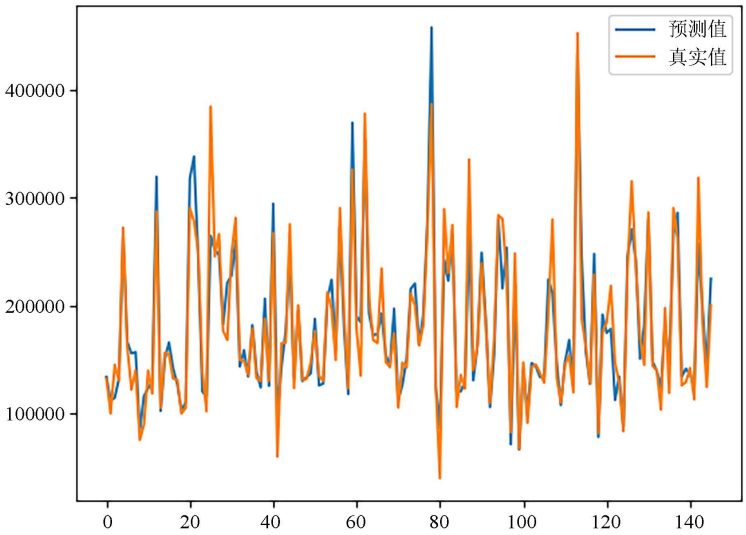

Figure 5. Prediction results are not improved
图 5. 未改进预测结果

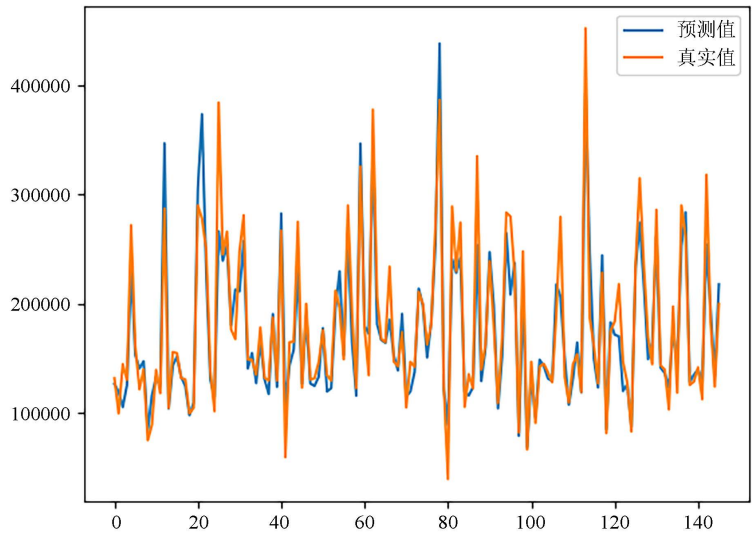

Figure 6. After the improvement, the parameter is not tuned to predict the result
图 6. 改进后未调参预测结果

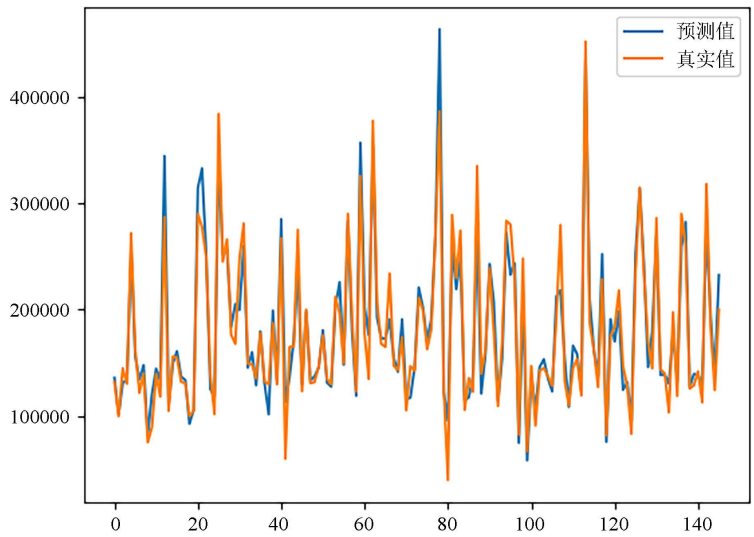

Figure 7. Predict the results after the improved parameter tuning
图 7. 改进后调参后预测结果

通过观察预测值与真实值之间的重合程度(见图 5，图 6，图 7)来把握模型的拟合程度，相比于原始模型，改进后的模型预测值与真实值之前的重合度更高。但这样观察明显不够精准，因此可以选择通过对比评价指标的大小来判断模型的质量高低，表 4 清晰地展示改进前后的 XGBoost 模型各个指标的对比结果。

一般来说，MAE、MSE、MAPE 的值越小，说明预测模型拥有更好的精确度，相反，的值越大，说明预测模型拥有更好的精确度。从表 4 可以看出，未改进的模型在预测房屋价格方面的精度较好，其值为 0.886，当未调整参数时，XGBoost 模型的值反而降到 0.864，因此我们需要对模型的参数进行调整，参数调整后可以达到 0.912，MAE 和 MSE 的值也显著下降，可以看出改进后的 XGBoost 模型比改进前

的模型具有更好的拟合效果。

Table 4. Comparison of model prediction accuracy
表 4. 模型预测精度对比

XGBoost 模型	未改进	改进后未调参	改进后调参后
MAE 值	16843.116	18085.025	15606.421
MSE 值	607465378.018	725304401.735	466409911.636
MAPE 值	0.097	0.101	0.096
R^2 值	0.886	0.864	0.912

采用 XGBoost 模型和改进后的 XGBoost 模型均有不错的预测效果，但改进后的 XGBoost 模型预测精度更高，有着更好的拟合效果。这说明改进后的 XGBoost 模型具有更好的学习能力，同时也说明了通过分数阶梯度下降算法来优化 XGBoost 模型是可行的。实验数据也表明了不是所有分数阶模型一定比整数阶模型效果好，XGBoost 模型中调参是很重要的一部分。

5. 总结与展望

本文首先将近几年关于分数阶梯度下降算法的发展与研究进行总结归纳，接着利用改进后的 XGBoost 模型对数据集进行实验，不同的模型参数模拟出的收敛结果不一样，需要选择合适的参数。在以往的研究中，没有学者研究过分数阶 XGBoost 模型，本课题在简化目标函数的过程中，将分数阶梯度下降与 XGBoost 模型相结合，并选择合适的分数阶，得到新的分数阶 XGBoost 模型。

本文有以下两个创新点：

1) 选择分数阶改进梯度下降算法对 XGBoost 模型进行优化。

2) 在数据集上将传统的 XGBoost 模型与改进后的模型的表现做了对比，得出改进后的模型具有更好的效果的结论。

虽然在现阶段取得了一定的研究成果，但仍存在一些不足和需要改进的地方，本文初步探究了基于分数阶梯度下降的 XGBoost 模型，实例分析部分仅探讨了预测问题，未考虑分类、排序等任务，下一阶段仍需研究其他类型问题，进一步加强模型鲁棒性，使其适用于复杂情形下的问题。实验表明 XGBoost 模型可以与分数阶微积分结合，分数阶梯度下降算法可以应用于一些其他模型，例如 GBDT、lightGBM 等，然后用于研究相关的优化问题，或者将其他的改进梯度下降算法应用于 XGBoost 模型。

参考文献

[1] 史加荣, 王丹, 尚凡华, 等. 随机梯度下降算法研究进展[J]. 自动化学报, 2021, 47(9): 2103-2119.

[2] 王书博. 基于改进梯度下降法求解多元线性回归方程[J]. 数学的实践与认识, 2022, 52(10): 167-172.

[3] 谢景伊. 基于时间分数阶梯度下降方法的神经网络训练方法[D]: [硕士学位论文]. 贵阳: 贵州大学, 2022.

[4] 许少华, 宋美玲, 许辰, 等. 一种基于混合误差梯度下降算法的过程神经网络训练[J]. 东北石油大学学报, 2014, 38(4): 92-96+11-12.

[5] Chen, Y., Gao, Q., Wei, Y. and Wang, Y. (2017) Study on Fractional Order Gradient Methods. *Applied Mathematics and Computation*, **314**, 310-321. https://doi.org/10.1016/j.amc.2017.07.023

[6] Pu, Y., Zhou, J., Zhang, Y., Zhang, N., Huang, G. and Siarry, P. (2015) Fractional Extreme Value Adaptive Training Method: Fractional Steepest Descent Approach. *IEEE Transactions on Neural Networks and Learning Systems*, **26**, 653-662. https://doi.org/10.1109/tnnls.2013.2286175

[7] Chaudhary, N.I., Zubair, S. and Raja, M.A.Z. (2017) A New Computing Approach for Power Signal Modeling Using Fractional Adaptive Algorithms. *ISA Transactions*, **68**, 189-202. https://doi.org/10.1016/j.isatra.2017.03.011

[8] Wang, Y., He, Y. and Zhu, Z. (2022) Study on Fast Speed Fractional Order Gradient Descent Method and Its Application in Neural Networks. *Neurocomputing*, **489**, 366-376. https://doi.org/10.1016/j.neucom.2022.02.034

[9] Wang, J., Wen, Y., Gou, Y., Ye, Z. and Chen, H. (2017) Fractional-Order Gradient Descent Learning of BP Neural Networks with Caputo Derivative. *Neural Networks*, **89**, 19-30. https://doi.org/10.1016/j.neunet.2017.02.007

[10] Khan, S., Naseem, I., Malik, M.A., Togneri, R. and Bennamoun, M. (2018) A Fractional Gradient Descent-Based RBF Neural Network. *Circuits, Systems, and Signal Processing*, **37**, 5311-5332. https://doi.org/10.1007/s00034-018-0835-3

[11] Chen, M., Chen, B., Zeng, G., Lu, K. and Chu, P. (2020) An Adaptive Fractional-Order BP Neural Network Based on Extremal Optimization for Handwritten Digits Recognition. *Neurocomputing*, **391**, 260-272. https://doi.org/10.1016/j.neucom.2018.10.090

[12] 阚涛, 高哲, 杨闯. 采用分数阶动量的卷积神经网络随机梯度下降法[J]. 模式识别与人工智能, 2020, 33(6): 5-9.

[13] 李兴怡, 岳洋. 梯度下降算法研究综述[J]. 软件工程, 2020, 23(2): 1-4.

[14] 刘伟. 基于改进的 XGBoost 的短期交通流预测[D]: [硕士学位论文]. 太原: 太原理工大学, 2020.

[15] Wang, Y., He, Y. and Zhu, Z. (2022) Study on Fast Speed Fractional Order Gradient Descent Method and Its Application in Neural Networks. *Neurocomputing*, **489**, 366-376. https://doi.org/10.1016/j.neucom.2022.02.034

[16] 李梦宇, 张泽亚, 张知, 等. 基于 CART 算法的电能表故障概率决策树分析[J]. 电力大数据, 2017, 20(10): 7-10+60.

[17] 陈玉全. 分数阶梯度下降法基础理论研究[D]: [博士学位论文]. 合肥: 中国科学技术大学, 2020.

[18] 原琨. 基于 XGBoost 的移动优惠券使用预测及影响因素分析[J]. 广西质量监督导报, 2018(12): 99-100.