2022 年中南大学数学建模竞赛

承　诺　书

我们完全清楚,在竞赛中必须合法合规地使用文献资料和软件工具,不能有任何侵犯知识产权的行为.否则我们将失去评奖资格,并可能受到严肃处理.

我们以中国大学生名誉和诚信郑重承诺,严格遵守竞赛章程和参赛规则,以保证竞赛的公正、公平性.如有违反竞赛章程和参赛规则的行为,我们将受到严肃处理.

我们授权中南大学数学建模竞赛指导教师团队,可将我们的论文以任何形式进行公开展示（包括进行网上公示,在书籍、期刊和其他媒体进行正式或非正式发表等）.

我们参赛选择的题号（从 A/B/C/D/E 中选择一项填写）：A

我们的报名参赛队号（7 位数字编号）：2022147

参赛队员 (电子签名) ：

1. 蒋昕妍

2. 文启哲

3. 方子贤

日期：2022 年 6 月 30 日

（请勿改动此页内容和格式. 以上内容请仔细核对,如填写错误,论文可能被取消评奖资格.）

基于梯度提升决策树的决策模型

摘　要

随着新材料研究领域的发展,如何通过成分含量准确检测材料性能问题是材料成分-性能映射关系中的重要研究课题.本文针对化学成分含量和力学性能相关性复杂的问题,基于把握主要成分思想，以建立成分性能之间复杂关系模型、提高决策效率为目标,建立了基于梯度提升决策树的回归模型,并结合神经网络设计出决策方法.

【化学成分和力学性能的非线性关系较为复杂，高效的决策预测模型对降低生产成本提高生产效率具有重要意义．本文针对化学成分和力学性能相关性决策效率低的问题,基于回归分析与机器学习算法,以建立化学成分和力学性能定量关系、简化模型复杂度和提高决策效率为目标,建立了基于梯度提升决策树的成分-性能回归模型，并结合神经网络实现降低实验成本提高决策效率的目的．】

针对问题一：对力学性能的变化情况进行分析,考虑到力学性能是各化学成分的多元非线性函数,该函数表达式未知.基于拟合的思想,假设该多元函数是若干组基本初等函数的组合,并用最小二乘法确定相关系数.使用残差、平均相对误差等指标衡量拟合模型的近似程度.

针对问题二：对于化学成分和力学性质之间的定量关系,考虑到他们之间的相关性复杂,考虑先找到对力学性能影响大的化学成分特征,于是采用了决策树模型.为了适应更一般的损失函数,我们采取基于梯度提升决策树模型 GBDT,并通过对损失函数性能的测试,选择性能最优的 huber 损失,取 70%数据进行模型训练,30%作为测试集,得到性能较好,准确度较高的模型.

针对问题三：对于模型的实际应用,考虑到以后进一步测量力学性质的实验需要,针对特殊力学性质的材料设计要求,我们设计了一种实验指导思想.通过机器学习设计出合适的模型,根据 GBDT 模型提供的特征重要性占比找出对于某种力学特性影响较大的化学成分,再通过 BP 神经网络的反向传播得到关于化学成分的预测值.基于该预测值,我们可以简化后续针对具有特殊力学性质材料的成分实验.

关键词：　多元非线性回归　决策树　梯度提升　神经网络　重要特征筛选

目录

1 问题综述

1.1 问题背景

材料是人类赖以生存和发展的物质基础,随着新技术革命浪潮的发展,材料革新对产业发展和技术进步有重要作用.材料性能与其组成成分关系密切,传统的化学冶金工艺中化学成分设计通常是根据经验或者试错,对于成分复杂的设计,很难掌握其对于工艺性能的影响调控规律,造成材料研发周期漫长,材料性能提升缓慢的问题.因此,要从根本上解决这个问题,就需要在对材料成分性质进行定性掌握的基础上,实现对各化学成分的定量化描述和设计.[2]

近年来,机器学习在构建化学成分与材料力学性能对应定量关系的研究得到发展,基于机器学习的大部分模型都能够建立成分-性能的映射关系,并提出相关改进算法如基于深度神经网络的数据驱动模型,能够高效快速的实现成分-性能之间的定量关系,但是实际准确性并不是很高；同时,在化学和材料领域中机器学习的数据集往往来源于实际生产中或者是相关书籍资料中,缺乏强大的数据库支撑,小样本进行机器学习训练使得预测值和实际值的差距较大,而当前解决该问题的经验采样以及贪婪采样等扩充数据集的方法,随着训练集的数量增加,虽然模型准确性会有所提高,但是他需要训练时间也会相应增长,得到结果速度慢.于是,我们提出了基于梯度提升决策树的成分-性能回归模型,在保证模型准确度的基础上,使得定量关系的转换更加迅速,实现高效快速的定量转换.

1.2 问题提出

在新材料研究的过程中,要求能够快速准确的实现化学成分和力学性能之间的定量关系,以达到减少实验成本提高评估效率的目的.其中,问题主要由温度,元素化合物稳定性,材料微观结构这几个要素构成,.通常情况下,化学原料在不同高温情况下会发生不同的化学反应,这就有可能造成最终相同的元素构成不同的混合物从而对力学性能造成影响； 并且,元素之间的活泼性高低不同,使得反应先后顺序也可能不同,这对于最终形成的材料内部结构具有很大的影响；前两点都有可能对工艺材料的微观结构造成极大影响,而本文对探讨化学成分含量与力学性质的关系进行探讨,所以需要基于上述分析控制单一变量进行研究.

需要从定性分析、定量分析、建立最佳决策模型的角度考虑, 解决以下 3 个问题:

(1) 问题 1: 给定化学成分和力学性能的数据,要求建立数学模型对它们之间的相关性进行分析,画出每个变量与各力学性质的相关性图像,并计算出相关系数矩阵.根据经验,不同化学成分之间与力学性能的相关性是一个复杂的问题,我们需要探究所有化学成分对于同一力学性能的影响,并且画出多个变量共同作用和力学性质的相关性拟合曲线.

(2) 问题 2: 在问题 1 的基础上,我们需要得到模型拟合的相关性系数,从而计算出表中给定的缺失测量数据的熔炼号产出材料的力学性能均值和标准差.

(3) 问题 3:在前两问的基础上,根据已有的化学成分和性能关系,要求设计实验以达到尽快为特定力学性能需求进行配方,达到用最小成本设计出符合性能的材料.结合实际,可以准确满足不同客户对于材料性能的需求,同时提高材料生产厂商利润,在科学研究方面快速准确的材料性能检测能够减少人力、物力等资源的消耗,能够有效推动新材料研究领域的发展.

2 模型假设与符号说明

2.1 模型基本假设

(1) 假设忽略材料力学性能受到温度电磁场等外部因素的影响
(2) 假设忽略测试用的材料几何结构不同对于力学性能测试结果的影响
(3) 假设忽略不同熔炼炉环境和高温下不同化学反应对材料微观结构造成的影响
(4) 假设几种化学成分共同对力学性能起较大的决定作用而且他们之间的关系是非线性的复杂关系,而单个化学成分对力学性能的影响不显著
(5) 假设化学成分对力学性能的影响函数是若干种基本初等函数的线性组合
(6) 由不同材料的力学性能在不同温度、不同压强下不同,因此假设力学性能数据是在相同环境条件下测量的

2.2 符号说明

本文定义了如下 6 个使用次数较多的符号：

表 1 符号说明

符号	含义	单位
Zscore(En)	标准化之后的化学成分指标	无
En	化学成分代号	无
ε	多元线性回归方程常数项	无
θ_m	经验风险最小化时参数	无
g_i	损失函数的一阶导数	无
h_i	损失函数的二阶导数	无

3 数据预处理

3.1 数据处理

3.1.1 数据清洗

由于提供的原始化学成分数据来源于生产工作或者是文献、书籍中,由于这些数据在不同系统的存储是混乱的,没有标准化管理,由于一些人为因素如检测计算错误、记录错误或者是其他因素如炼制环境的异常改变使得当下测得的采用数据出现异常或者缺失,我们得到的数据是杂乱无章的,因此需要进行数据清洗.

在分析数据时候,我们发现化学成分采样样本和熔炼号对应的力学性能都存在缺失情况,为了保证数据集大小足够,我们采用插值填充的方法补全缺失值.同时,样本中存在一些与其他样本偏差较大的值,我们考虑是在测量统计时发生的记录错误或者是测量仪器异常导致的,因此为了提高数据集的质量,除去噪声矛盾数据,我们选取异常点附近数据的平均值代替该异常数据.

对于重复数据,其对应的属性能够由另外和他重复的数据导出,因此我们通过筛选删除冗余数据.

3.1.2 数据规约

由于每个熔炼号基本都进行两次采样,因此我们对相同熔炼号不同采样得到的化学成分数据进行取平均值数据集成同样,在力学性能数据中,同一个熔炼号炼制的材料对应了多组力学性能数据,我们同样将相同熔炼号对应的力学性能取平均值.从而形成了化学成分、熔炼号、力学性能三者之间的联系,在保证原始数据完整性的前提下减少了数据规模,使用规约后的数据集进行求解分析更加有效,降低了数据的冗余度.

分析化学成分是一个 6 维的数据,我们考虑是否能通过主成分分析的方法找出具有代表性的主要成分,从而对这一组熔炼炉中炼制的材料的化学成分投入有一定的掌握,我们使用 SPSS 对六中化学成分进行主成分分析,结果如下：

表 2 成分矩阵

成分矩阵[a]				
	原始		重新标度	
	成分		成分	
	1	2	1	2
Zscore(E1)	0.851	0.074	0.851	0.074
Zscore(E2)	0.450	-0.582	0.450	-0.582
Zscore(E3)	0.690	0.332	0.690	0.332
Zscore(E4)	0.797	0.255	0.797	0.255
Zscore(E5)	0.080	0.666	0.080	0.666
Zscore(E6)	-0.460	0.623	-0.460	0.623

提取方法：主成分分析法.

a. 提取了 2 个成分.

由表可以看出,提取了两个主成分,其中主成分1代表E1、E3、E4,主成分2代表E2、E5、E6.可以看出在这一系列熔炼号中投放各种化学成分的含量是彼此有联系,而不是随机投放,找出生产这类材料的规律,成分1、3、4的投入含量往往一起增加一起减少,同理成分2、5、6的投放也具有相同的规律.

4 问题分析

4.1 问题分析

题目以材料化学成分对性能的影响为背景, 介绍了化学组分比例对力学性能的影响, 问题一要求我们定性描述化学成分对力学性能的影响, 问题二要求我们在问题一的

基础上进一步定量研究成分对力学性能的定量影响.问题三要求我们在问题一二的基础上,设计符合性能要求从材料.

根据实际化工生产过程中,温度环境的不同造成产生不同的化学反应,而最终形成材料混合物结构对于力学性能的影响较大,本文只探讨化学成分的影响,因此应该尽量避免微观结构的不同造成的影响.因此在选取成分检测数据时,要控制不同熔炼炉的熔炼环境相同.

结合实际,模型重点在与提高化学成分含量和力学性能之间对应关系预测的准确性.我们需要考虑采样误差和熔炼环境对于模型结果的影响,因此,为了评价模型的性能,我们提出了模型准确度、高效性、稳定性等评价指标.

4.1.1 问题一分析:

问题一要求我们对不同化学成分与材料各种力学性质进行相关性分析.根据表格中的数据我们不妨忽略其他因素对于力学性质的影响,由于各熔炼号都进行两次成分化验,同时相同熔炼号对应多个力学性能测试值,这就要求我们进行平均化处理,从而构建出相关性模型得到各化学成分和力学性能之间的相关系数.

基于给出的有关实验数据, 我们首先考虑一般情况下各个变量独立地对力学性能产生影响.而单变量相关性的分析结果表明,变量对力学性能独立影响的水平较弱,查找相关资料发现,各个化学成分之间的共同作用于力学性质,要求建立多元线性回归方程进行拟合,考虑力学性能是多个自变量耦合的复杂非线性函数.

4.1.2 问题二分析:

问题二是在问题一的基础上,要求我们确定缺失真实数据的熔炼号中产出材料的力学性能均值和标准差.因此,这需要我们确定化学成分和力学性能之间的定量关系,可以沿用问题一中求解的多自变量耦合的非线性函数进行求解.

同时由于模型参数较多复杂度太高,同时求平均后的数据集大小减少,非线性函数的准确度和精度很难提高.所以,我们考虑使用机器学习的方法,找到一种适用于处理非线性低维数据,且对于异常值的鲁棒性较高的模型来拟合非线性关系

4.1.3 问题三分析:

要达到用最小成本设计出符合性能要求的材料,既要求我们既达到性能指标,又要减少实验次数消耗.即需要我们找到一个需要训练次数少且准确度高的模型.训练次数少,就要求每一次训练都尽可能高效,设计材料化学成分实验,需要我们在掌握了各成分与力学性能定性关系的基础上,筛选出可能目标解,合理规划每次选择实验的化学成分含量,使得每次查找都不盲目,尽可能朝着最优解的方向进行.

问题的关键在于我们要充分掌握化学成分含量和力学特性之间的关系,明确影响关系后,建立最佳实验决策模型.

5 模型建立与求解

5.1 问题一模型建立与求解

5.1.1 偏相关分析模型建立

由于化学成分和力学特性之间的相关关系是很复杂的,力学性质往往由多种化学成分共同决定,于是构建偏相关分析模型,在对某化学成分与某种力学性质的相关性进行分析时控制化学成分的影响,计算出偏相关系数.

5.1.2 偏相关参数确定

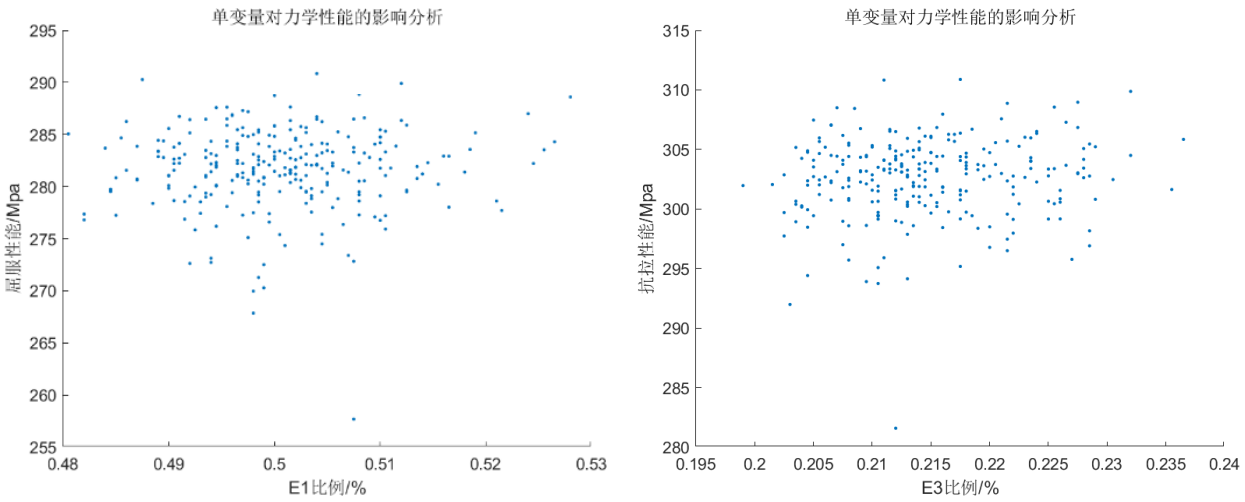

图 1 单变量对力学性能的影响分析

图中列举出 18 个关系图中的两个,E1 含量和屈服性能、E3 含量和抗拉性能之间的关系图,其他关系图中各化学成分和力学性质也没有表现出明显的相关性.由图可知,单变量和力学性能并没有显著的相关性.

利用 matlab 求出相关系数矩阵

表 3 相关系数矩阵

	屈服	抗拉	延伸率
E1	0.0424	0.0489	−0.1733
E2	0.0486	0.0197	0.0744
E3	0.1151	0.1122	−0.0239
E4	0.1347	0.1551	−0.1345
E5	−0.0004	0.0368	0.0387
E6	−0.1155	−0.068	0.0287

同时,使用 SPSS 的偏相关系数求解功能进行求解,得到相同的相关系数.从相关系数来看,单个化学成分对于力学性能的决定性作用较小,于是得出结论各化学成分是共同作用影响力学性能的.

5.1.3 多元线性回归方程的建立

从实际情况分析,材料的力学特性和多种化学成分有关,成分和力学特性的并不是简单的一一对应的线性关系.于是,尝试找到力学性能受各化学成分含量的函数关系.根据 E1~E6 六个化学成分含量和 W1~W3 三个力学性能指标构建多元线性回归方程,有

$$\begin{pmatrix} W1 \\ W2 \\ W3 \end{pmatrix} = \begin{pmatrix} a_{11} & \dots a_{16} & a_{17} \\ \vdots & \ddots & \vdots \\ a_{31} & \cdots a_{36} & a_{37} \end{pmatrix} \begin{pmatrix} E1 \\ E2 \\ E3 \\ E4 \\ E5 \\ E6 \\ 1 \end{pmatrix} \tag{1}$$

记 $\beta_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \end{pmatrix}, \beta_2 = \begin{pmatrix} a_{12} \\ a_{22} \\ a_{32} \end{pmatrix} \cdots\cdots \varepsilon = \begin{pmatrix} a_{17} \\ a_{27} \\ a_{37} \end{pmatrix}, n \in N, n > 1,$ 则多元线性回归方程为 $Y = \varepsilon + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_6 X_6$

5.1.4 多元线性方程参数的确定

利用 matlab 编程,使用多元线性回归模型进行拟合,得到三组回归系数填入表中

表 4 屈服强度系数

E1	E2	E3	E4	E5	E6	W0
-47.1655	7.971919	44.51652	161.439	-12.2558	-175.381	270.5296

表 5 抗拉强度系数

E1	E2	E3	E4	E5	E6	W0
-37.111	3.216	32.277	158.826	18.554	-85.455	285.44

表 6 伸长率系数

E1	E2	E3	E4	E5	E6	W0
-8.254	4.461	2.492	-5.015	4.715	5.421	10.877

由相关性系数可以分析得出,E1、E5、E6 三种化学成分含量与材料屈服强度呈现负相关影响,其余几种化学成分含量与屈服强度呈现正相关关系.E1、E6 成分含量与材料抗拉强度呈现负相关关系,其余化学成分含量对拉康强度起增强作用.E1、E4 对材料伸长率有削减作用,其余成分含量的增加可促进伸长系数的增加.

我们以熔炼炉编号递增顺序为横坐标,依次画出各组线性回归模型对于样本的拟合图像.如图可知,模型对于样本的拟合程度不太高,较多的点的偏差较大.

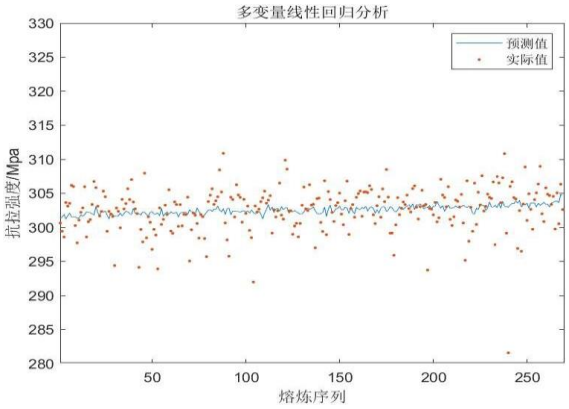

图 2 拉抗强度回归分析

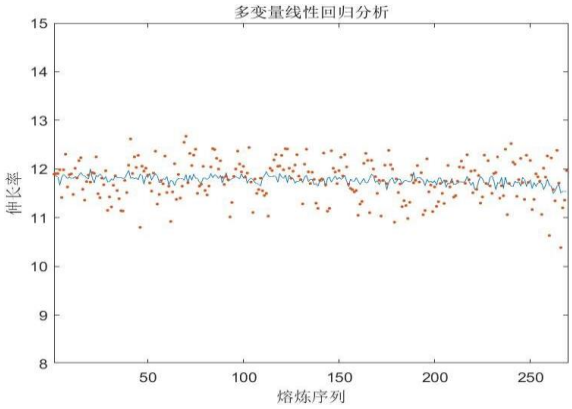

图 3 延伸率回归分析

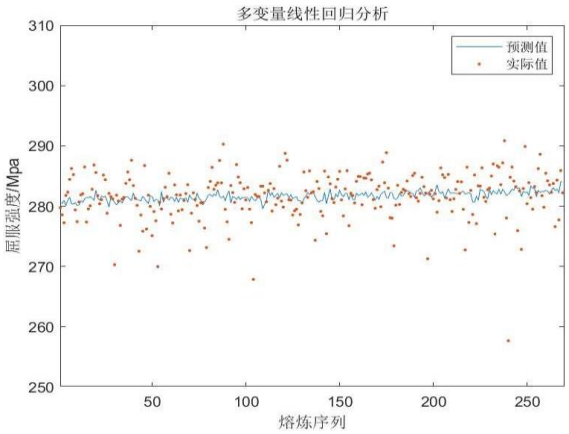

图 4 屈服强度回归分析

5.1.5 非线性模型建立和求解检验

由以上分析可知多元线性回归模型的拟合程度不高,依据我们的模型假设,我们构建出非线性模型并增加模型复杂度,以此采用了抛物线拟合、抛物线与三角组合、抛物线与三角与对数组合、抛物线与三角与对数与指数组合多种非线性模型进行拟合,得到各种模型拟合的相对误差.

其中抛物线拟合：

$$Y = a_1X_1 + a_2X_2^2 + \cdots a_6X_6^6 + a_7X_1X_2 + a_8X_1X_3 + a_9X_1X_4 + a_{10}X_1X_5 + a_{11}X_1X_6 + \\ + \cdots a_{21}X_5X_6 + a_{22}X_1 + a_{23}X_2 + \cdots a_{27}X_0 + a_{28} \tag{2}$$

三角拟合：

$$Y = \sum_{i=1}^{6} a_i \sin_i \left(b_i x_i\right) \tag{3}$$

指数拟合：

$$Y=\sum_{i=1}^{6}a_i e^{b_i x_i} \tag{4}$$

对数拟合：

$$Y=\sum_{i=1}^{6}a_i \ln x_i \tag{5}$$

$$Y=\sum_{i=1}^{6}ax_i+\mathbf{x}^T B\mathbf{x}+\sum_{i=1}^{6}c_i \sin(d_i x_i)+\sum_{i=1}^{6}e_i \exp\{f_i x_i\}+\sum_{i=1}^{6}g_i \ln x_i+\varepsilon \tag{6}$$

其中 $B=\begin{pmatrix} b_{11} & \cdots & b_{16} \\ \vdots & \ddots & \vdots \\ b_{61} & \cdots & b_{66} \end{pmatrix}, X=\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}$

表 7 各模型对力学性能拟合相对误差

	抛物线	抛物线与三角	抛物线与三角与对数	抛物线与三角与对数与指数
屈服强度	0.0095	0.0093	0.0093	0.0093
伸长率	0.0076	0.0074	0.0074	0.0074
拉抗强度	0.0245	0.0244	0.0241	0.0241

由于化学成分在冶炼时因为各成分含量的比例不同或者含量多少不同,对于构成材料的微观结构具有不同的影响,从而对力学性能产生复杂的影响.这也证实了我们假设中他们之间是非线性相关的结论正确.

通过进一步查找资料,我们发现不同晶体结构的金属材料在温度变化下屈服强度变化趋势不同,其化学成分不同材料内部晶粒尺寸大小和均匀性都会对延伸率造成影响,化学结构中的级性基团和取代基都会对材料的拉伸强度造成影响.这些因素对力学性能的影响都是非线性的,因此,化学成分不不仅自身的力学性质有差异,他们也通过组成材料的围观物理结构不同,从而间接的影响力学性能.

5.2 问题二模型建立与求解

5.2.1 GBDT 梯度提升决策树模型简介

首先介绍提升树算法,第一步即对提升树进行初始化 $f_0(x)=0$,之后第 m 步的模型为

$$f_m(x)=f_{m-1}(x)+T(x;\theta_m) \tag{7}$$

其中, $f_{m-1}(x)$ 表示前一步模型,我们的目标事寻求经验风险最小化时的 θ_m ,

$$\theta_m=\arg\min_{\theta_m}\sum_{i=1}^{n}L(y_i,\ f_{m-1}(x_i)\ +T(xi;\theta_m)) \tag{8}$$

在使用提升树解决回归问题时,为了适应更一般的损失函数进行改进采取梯度提升树 ,其本质是利用最快梯度下降法,采用损失函数的负梯度作为改进的提升树回归算法的近似残差,从而对一个回归树进行拟合.梯度表示为：

$$-[\frac{\partial L(y,\ f(xi))}{\partial f(xi)}]f(x)=f_{m-1}(x) \tag{9}$$

GBDT (Gradient Boosting Decision Tree) 梯度提升迭代决策树算法,是一种将决策树作为基学习器的 boosting 算法.他与常见的 boosting 算法——AdaBoost 算法不同的是,它要求的弱学习器必须使用 CART 决策树模型,而且要保证损失函数尽可能要小,这就使得 GBDT 算法兼得了 boosting 算法和决策树算法各自的优点,一定程度上解决了单棵决策树容易过拟合的缺点的同时,保留了决策树对特征友好,不用做额外的特征工程的优点,从而减少了时间消耗.

5.2.2 基于梯度提升决策树的成分-性能回归模型

基于本题的特征数据——材料的化学成分,维度不高,并且没有呈现明显的线性关系,且对于得到力学性能的结果有一定要求,我们尝试采用机器学习模型来拟合这个多变量的回归问题.提出以 GBDT 为基础,通过改进损失函数的方法来获得最优模型求得最优解.

梯度提升算法模型主体如下：

输入：我们以化学成分含量为自变量,以力学性能为因变量,选取实验数据中 70%的数据作为训练集 $T=\{(x_1,y_1),(x_2,y_2),...,((x_N,y_N)\},x_i\in R^n,y_i\in R$,30%的数据作为测试集进行训练,选择合适的损失函数 $L(y,f(x))$,拟合出最佳目标函数

输出：回归树 $\hat{f}(x)$ 即我们最终的目标模型

1：初始化 $f_0(x)=\arg\min_c\sum_{i=1}^{n}L(y_i,c)$

2： $for\ m=1,2,\ldots M\ do$

(a)按照下面公式计算每个训练集样本的残差

$$r_{mi}=-[\frac{\partial L(y_i,f(x_i))}{\partial f_{m-1}(x_i)}],i=1,2\ldots N \tag{10}$$

(b)拟合残差学习一个回归树,得到第 m 棵树的叶节点区域

$$R_{mj},\ j=1,2\ldots,J \tag{11}$$

(c)在通过拟合残差学习到的回归树中找出一颗误差最小的树

$$c_{mj}=\arg\min_{c}\sum_{x_j\in R_{mj}}L(y_i,f_{m-1}(x_j)+c) \tag{12}$$

3：得到回归问题目标提升树

$$\hat{f}(x)=f(x)=\sum_{m=1}^{M}\sum_{j=1}^{J}c_{mj}I(x\in R_{mj}) \tag{13}$$

下图为 GBDT 梯度提升决策树原理流程图：

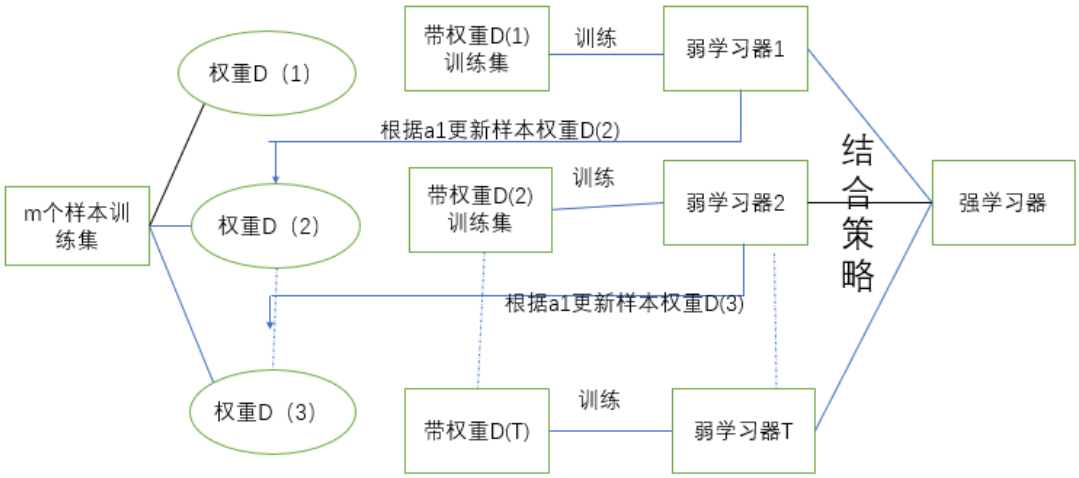

图 5　GBDT 梯度提升决策树原理图

5.2.3 损失函数的选取

基于 GBDT 的回归模型使用的常见损失函数有 huber 损失、平方损失、均方差损失,我们尝试通过对比不同损失函数对于该模型的回归拟合性能,对 GBDT 进行模型性能的优化.

选取平方损失（ls）作为损失函数,则目标函数可化简为

$$= \sum_{i=1}^{n} l(y_i, T_{t-1}(x_i;\theta_m) + f_{t-1}(x_i)) + \Omega(f_t) + \text{ constant} \tag{14}$$

$$= \sum_{i=1}^{n} (y_i - (T_{t-1}(x_i;\theta_m) + f_{t-1}(x_i)))^2 + \Omega(f_t) \tag{15}$$

再利用泰勒公式将损失函数展开,化为

$$Obj^{(t)} \approx \sum_{i=1}^{n} \left[g_i T_{t-1}(x_i;\theta_m) + \frac{1}{2} h_i T_{t-1}(x_i;\theta_m) \right] + \Omega(f_t) \tag{16}$$

其中 g_i 是损失函数的一阶导数 h_i 是损失函数的二阶导数.

选取 huber 函数作为损失函数,利用决策树作为基学习器,在迭代的过程中不断优化 huber 函数,同样通过求导得到最优化模型,利用最小化损失函数策略完善目标函数.

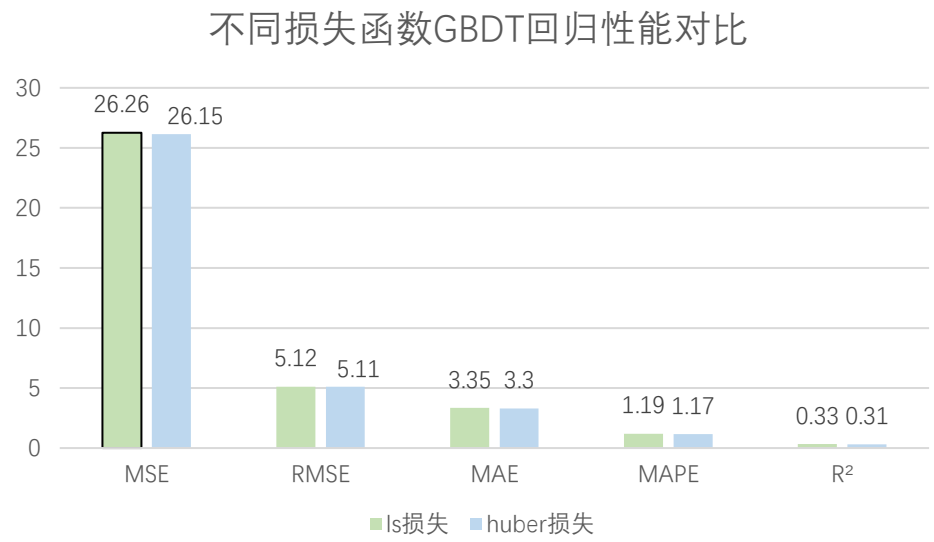

图 6 不同损失函数 GBDT 回归性能对比

5.2.4 模型问题求解

我们采用 SVR、BP 神经网络、GBDT 三种模型分别对力学性能进行拟合.将 70%的化学成分数据和对应的力学性能数据输入模型进行训练.对三种模型的回归拟合优度进行评价结果,有表 6 得出 GBDT 模型的回归拟合性能相比于其他模型更好.

选用 GBDT 模型计算出各熔炼号力学性能指标如表 8：

表 8 各个熔炼号力学性能指标

熔炼号	均值			标准差		
	屈服	抗拉	延伸率	屈服	抗拉	延伸率

90624	279.8333	301.1538	11.81472	4.183275	3.757993	0.645807
90626	281.7062	302.721	11.85602	3.997555	3.602548	0.671521
90627	281.4009	302.2163	11.88066	4.420585	3.902905	0.69017
90628	282.0643	302.7823	11.90026	4.408762	3.907243	0.698124
90629	281.0238	301.8968	11.81566	4.298067	3.809072	0.67472
90630	281.8326	302.6225	11.75769	4.461338	3.908738	0.696649
90631	280.4758	301.5745	11.67689	3.996573	3.371357	0.678676

5.3 问题三模型建立与求解

5.3.1 梯度提升决策树模型优势介绍

决策树求解决策结果的过程就是通过递归,不断寻求一个最优划分属性的过程.而这个最优划分属性就是决策树选出的重要特征.因此,决策树天生就具有寻找重要特征的性质.

基于第二问中建立的梯度提升决策树模型,在决策过程中能得出重要特征,这些重要特征对应节点可以构造出的决策模型能够降低信息熵,使得决策结果更加准确.选取的这些特征,即是对决策结果有决定性作用的.即某些化学成分变量对于力学性能的改变其决定作用,这些化学成分变量相当于决策树的子节点,能够将力学性能相差较大的样本分离开来.

5.3.2 实验模型设计

决策树模型得到的每一种力学性能所对应的重要特征,可以作为设计实验的重要参考.

对于机器学习模型减少成本相当于减少模型训练的时间和次数,同时保证决策准确性,只需要根据力学性能需求,找到其对应的重要特征.根据第二问决策树模型给出的影响每种力学性能的重要特征,我们利用神经网络的反向传播构建一个性能-成分的高效对应模型.

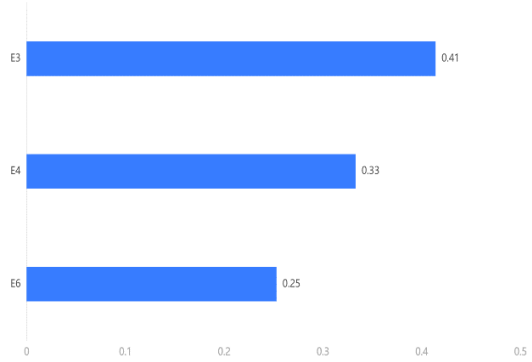

图 7 屈服性重要成分占比

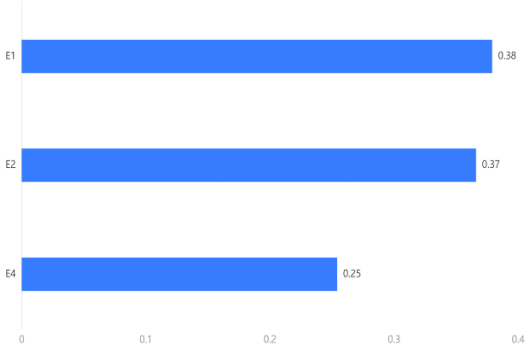

图 8 延伸性重要成分占比

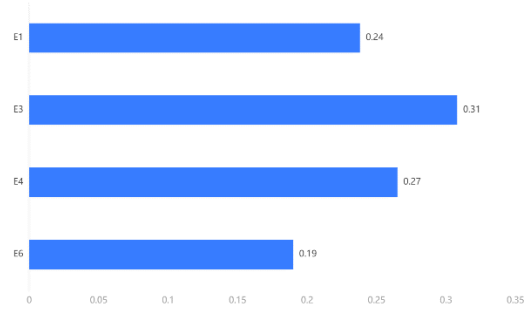

图 9 抗拉性重要成分占比

可以发现,在这个使用这个模型拟合材料屈服力学性质的时候,E3 这个化学成分对模型的影响较大,我们可以着重考虑 E3 这个化学成分对力学性质的影响,简化了实验的前期工作.模型可以将某一力学特性与某一化学成分的联系紧密程度反映出来,我们可以通过联系紧密程度的比较去选择相应的化学成分进行进一步的研究,这能让我们更有目的性地进行实验的设计.

神经网络反向传播原理如下：

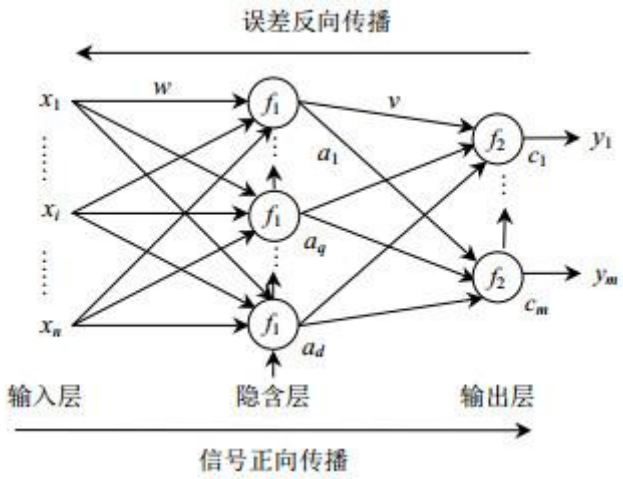

图 10 神经网络原理图

通过给出力学性能,通过反向传播计算修正出它和由 GBDT 选择出来的重要特征之间的权值关系,只研究力学性能和影响其主要成分之间的定量关系,得到的权值可能对应于多组重要化学成分特征的解,但是这相比于盲目调整各个化学成分或者是依据经验调整各化学成分配比来找到满足性能的一组解要更快,只需要在满足权值对应关系的一组解中去寻找,这样可以更加快速有效的找到性能对应的化学成分配比,减少了寻找对应成分的实验次数.构建出一个基于神经网络和 GBDT 重要特征筛选的决策模型.

在此模型基础上进行化学成分实验,每次实验都在模型给出的一组解中,选出一组化学成分含量进行验证并且在这组解的基础上进行含量的微调,得到更加精确的一组化学成分含量的解.同时,如果需求较复杂,例如需要同时满足多种力学性质的要求,我们可以通过对每一种力学性质对应解的集合取交集,缩小目标化学成分含量配比可能的范围,使得实验次数减少从而减少成本开销.

6 模型分析和检验

6.1 模型效果分析

下面是 GBDT 模型使用效果,由图 7 可知,模型的效果较好,对于屈服性能的预测和真实值的变化趋势大概一致,大部分预测值相对于真实值较高.我们发现模型在屈服性 280 到 285 之间的力学性能预测较好.由图 8 可知,抗拉性能的预测效果和屈服性能相近,同样存在部分预测值和真实值偏差较大,在抗拉性 200 到 305 之间的预测准确性较高.由图 9 可知,延伸性能的模型效果不如前两者.所以在实际应用中,若需要对应力学性能处于某个区间内的材料,我们可以采用此模型以达到较高的预测准确度.

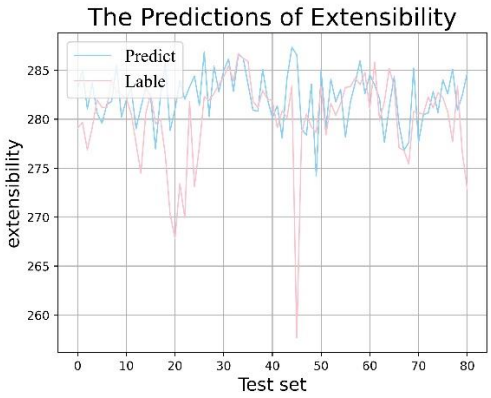

图 11 屈服性能预测值和标签值对比

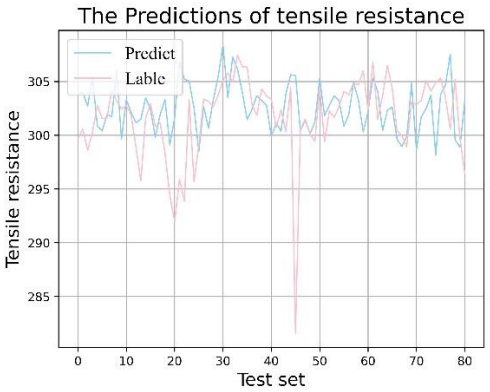

图 12 抗拉性能预测值和标签值对比

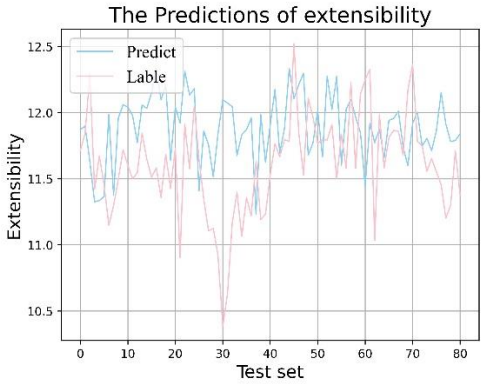

图 13 延伸性能预测值和标签值对比

6.2 模型误差分析

经过不断调整参数,改变损失函数,通过多次迭代,求得每一步要学习的函数,一次累加即可得到最终的模型；通过回归评价指标得出,相较于参数过多的 BP 神经网络和 SVR 算法,使用 GBDT 模型能够得到更好的预测结果.下面是屈服性回归分析性能分析

表 9 屈服性回归分析性能

	MSE			RMSE		
	SVR	BP	GBDT	SVR	BP	GBDT
训练集	17.318	14.537	0.067	4.161	3.813	0.259
测试集	9.228	14.76	14.76	3.038	3.842	5.315

	MAE			R^2		
	SVR	BP	GBDT	SVR	BP	GBDT
训练集	2.884	2.652	0.035	0.009	0.034	0.994
测试集	2.292	2.907	3.411	0.038	0.007	0.525

我们对各力学特性的 GBDT 回归参数性能进行对比,以基学习器个数为自变量,R^2 作为评价标准,讨论在不同放回采样率下的参数性能.发现随着基学习器的个数增多,参数性能会随之增加,于是我们可以通过增加基学习器的个数来提升 GBDT 的性能.同时,在通过在化学成分数据集中进行有放回采样,相当于增加了数据集的大小,使得模型性能得到优化.

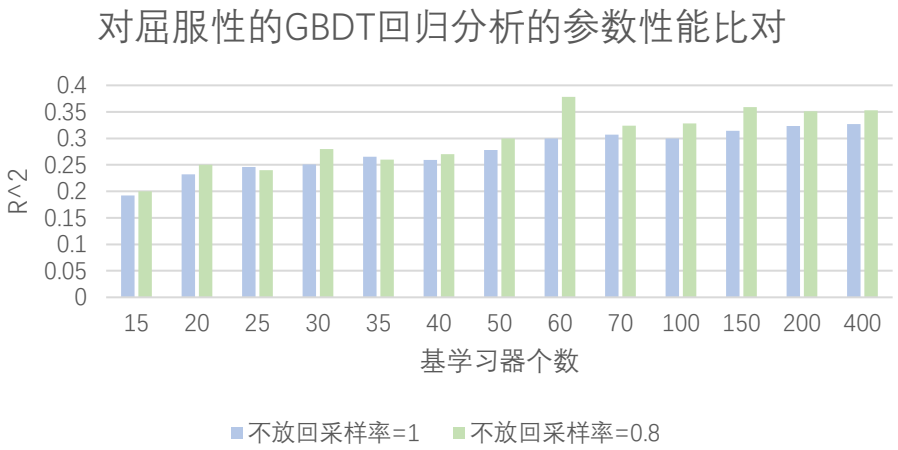

图 14 屈服性的 GBDT 回归分析参数性能比对

6.3 GBDT 模型稳定性分析

在实际生产应用过程中,由于对化学成分的取样多少以及熔炼炉中高温液体不同位置可能浓度不同,由于这些不可控因素的存在,取样成分含量并不能准确表示该熔炼炉中个化学成分的含量,实际熔炼炉中化学成分含量会在一个范围内波动.因此,需要检验基于 GBDT 模型的成分-性能回归模型对于测量精度扰动的控制效果.

假设每种化学元素的实际含量可能在测量含量的上下 5%范围内浮动,即

$$n_{real} = n_{std} * (1+e), e \sim U(-0.05, 0.05) \tag{17}$$

其中 n_{real} 表示真实化学成分含量, n_{std} 表示测量测量的化学成分含量.

建立在 GBDT 模型可以输出特征重要性的基础上,我们针对重要特征进行稳定性分析,使得屈服性的重要特征 E3、E4、E6,延伸性的重要特征 E1、E2、E4,抗拉性的重要特征 E1、E3、E4、E6,在测量含量上下 5%的范围内波动,各稳定性图像如下：

表 10 各成分对屈服强度的稳定性结果

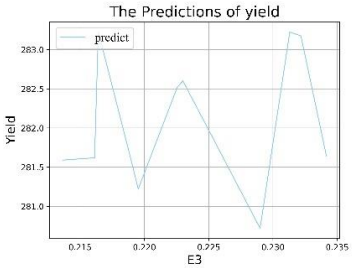

图 15 E3 对屈服性的稳定性

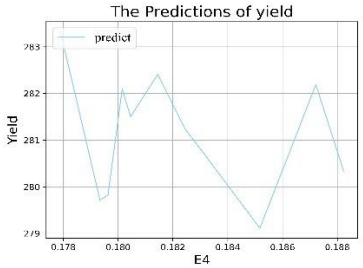

图 16 E4 对屈服性的稳定性

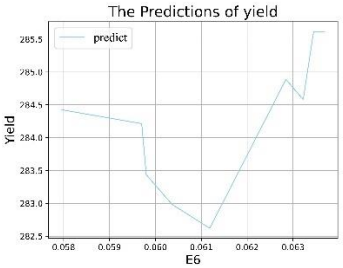

图 17 E6 对屈服性的稳定性

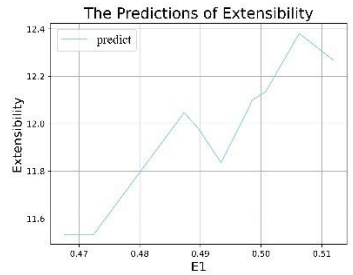

图 18 E1 对延伸性的稳定性

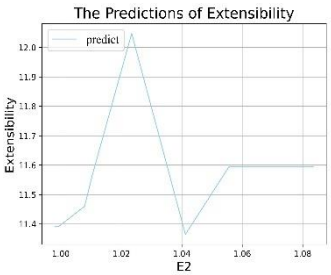

图 19 E2 对延伸性的稳定性

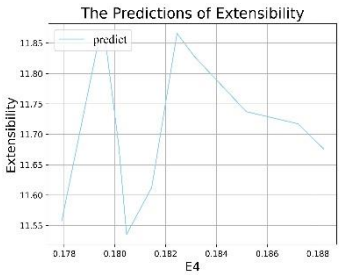

图 20 E4 对延伸性的稳定性

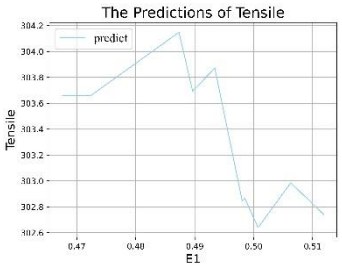

图 21 E1 对抗拉性的稳定性

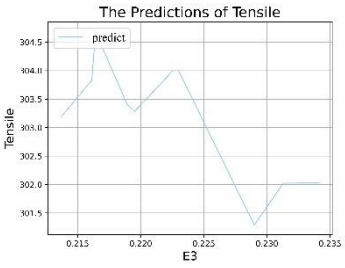

图 22 E3 对抗拉性的稳定性

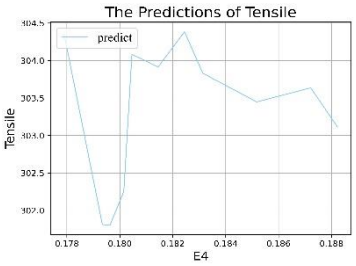

图 23 E4 对抗拉性的稳定性

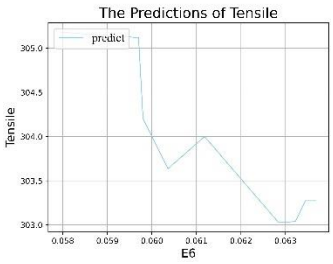

图 24 E6 对抗拉性的稳定性

由于纵坐标刻度间隔选取较小,从图像看来变化很大,但是在测量值上下 5%左右浮动对力学性能的误差控制在 1 左右,通过查找相关材料合格允许误差范围,该模型稳定性完全能够满足对精度要求不高的需求.

7 模型评价与推广

7.1 模型的优点

(1) 非线性线性回归模型充分结合实际, 考虑了复杂情况, 考虑多种可能的函数类型, 使拟合结果更接近真实值

(2) 采用梯度提升决策树模型,具有在计算阶段计算速度快,不同决策树之间可以并行计算的特点,相对于其他机器学习方法如 BP 和 SVR 可以对得到多元非线性问题中

影响力学性能的重要特征,从而减少特征的分析,进而加快计算速度,减少不重要因素对于决策的影响.

(3) 本文设计的实验是基于神经网络和 GBDT 重要特征筛选的决策模型的.GBDT 的重要特征选择功能减少了神经网络节点个数,使得神经网络更加简单,求解速度也有所提高.

7.2 模型的不足

(4) 由于时间问题,我们没有对该模型只适用于这六种化学成分对材料力学性质进行分析预测,对于含有其他元素或者缺失某种模型中规定的元素都无法得到准确的预测,因此该模型适用范围较窄,只对于某种或者某几种性质类似的材料具有准确分析的能力.

(5) 模型的稳定性不高,由于采样数据的测量值和实际值之间存在偏差导致轻微化学成分含量的变化就会造成损失.本模型将损失大概控制在 0~1 之间,这对材料科学研究领域来说稳定性是很差的,模型的稳定性直接与决策结果准确性和精确度挂钩,这不利于材料科学的发展

(6) 问题一采用的多种非线性模型复杂度提高,参数增加使得模型求解困难,对于数据集大小要求增加,甚至会导致模型过拟合的现象

(7) 模型扩展性低,基于 GBDT 的模型适合处理低维数据,而实际工艺流程中化学元素的量远远不止六个,当需要寻求大量化学成分含量和力学性能之间的映射关系时,模型算法的计算复杂度急剧增大,时间消耗的增加导致模型整体性能的衰减,不能满足实际用户的时间需求.

(8) 基于神经网络和 GBDT 特征筛选的模型中,我们只能对每种力学性能对应的重要特征含量进行合理的决策,但是其余非重要特征的具体含量模型无法得到,所以在确定重要特征的之后,还需要人工对非重要特征的含量进行规化,使得实验次数增加成本

参考文献

[1] 杨大地.多项式拟合中正规方程组的病态分析和改善[J].重庆大学学报(自然科学版),1993(03):104-111.

[2] 赵婉辰,郑晨,肖斌,刘行,刘璐,余童昕,刘艳洁,董自强,刘轶,周策,吴洪盛,路宝坤.基于Bayesian 采样主动机器学习模型的 6061 铝合金成分精细优化[J].金属学报,2021,57(06):797-810.

[3] 文成. 基于机器学习的高熵合金成分设计与性能优化[D].北京科技大学,2022.DOI:10.26945/d.cnki.gbjku.2022.000147.

附　录

附录 A：支撑材料列表

支撑材料列表

序号	文件名	材料说明
1	huitu.py	绘制稳定性图像
2	S.py	整理出包含均值和标准差的数据
3	SPSS_qufu.py	使用 SPSS 得到的材料屈服模型
4	SPSS_kanla.py	使用 SPSS 得到的材料抗拉性模型
5	SPSS_yanshen.py	使用 SPSS 得到的材料延伸性模型
6	qufu_spssReport.docx	SPSS 对材料屈服性模型的报告
7	kanla_spssReport.docx	SPSS 对材料抗拉性模型的报告
8	YANSHEN_spssReport.docx	SPSS 对材料延伸性模型的报告
9	稳定度 test.py	生成测试稳定度的数据
10	new.xlsx	数据集,用于模型的训练
11	E.mat	预处理后的化学成分
12	Y.mat	预处理后对应的力学性能

13	TEST.m	需要估计的熔炉的化学成分
14	fitting.m	拟合主程序,共编写了五个模型,第一问的分析基于线性模型.程序运行自动导入E.mat、Y.mat,并计算出模型相关待定系数
15	TEST_T2.m	预测力学性能.运行拟合主程序后运行,使用精度最高的模型预测 TEST.mat 中各熔炉材料对应的力学性能,并将结果保存到表格.

附录 B: 主要程序/关键代码

代码环	操作系统: macOS Mojave (Version 10.14.3) 编程语言: Python 3.7.1 (Anaconda Navigator 1.9.2)、matlab 编辑器: PyCharm 2018.3.2 (Professional Edition)、matlab R2019a

代码清单 1 计算拟合模型待定参数模块

```
    % 本程序用于计算拟合模型的待定参数
clear;
load Y;load E;
sumM=5;  % 使用的模型总数
tmp1=ones(1,7);
model_1=@(b,X)b(1)*X(:,1)+b(2)*X(:,2)+b(3)*X(:,3)+b(4)*X(:,4)+b(5)*X(:,5)+b(6)
*X(:,6)+b(7); %一次项

tmp2=zeros(1,28); % 含抛物线拟合
model_2=@(b,X)b(1)*X(:,1).^2+b(2)*X(:,2).^2+b(3)*X(:,3).^2+b(4)*X(:,4).^2+b(5)
*X(:,5).^2+b(6)*X(:,6).^2+...%平方项

b(7)*X(:,1)+b(8)*X(:,2)+b(9)*X(:,3)+b(10)*X(:,4)+b(11)*X(:,5)+b(12)*X(:,6)+...
%一次项
```

```
b(13)*X(:,1).*X(:,2)+b(14)*X(:,1).*X(:,3)+b(15)*X(:,1).*X(:,4)+b(16)*X(:,1).*X
(:,5)+... %交叉项

b(17)*X(:,1).*X(:,6)+b(18)*X(:,2).*X(:,3)+b(19)*X(:,2).*X(:,4)+b(20)*X(:,2).*X
(:,5)+...

b(21)*X(:,2).*X(:,6)+b(22)*X(:,3).*X(:,4)+b(23)*X(:,3).*X(:,5)+b(24)*X(:,3).*X
(:,6)+...
    b(25)*X(:,4).*X(:,5)+b(26)*X(:,4).*X(:,6)+b(27)*X(:,5).*X(:,6)+...
    b(28); %常数项

tmp3=unifrnd (1,100,1,34); %含三角函数拟合
model_3=@(b,X)b(1)*X(:,1).^2+b(2)*X(:,2).^2+b(3)*X(:,3).^2+b(4)*X(:,4).^2+b(5)
*X(:,5).^2+b(6)*X(:,6).^2+...%平方项

b(7)*X(:,1)+b(8)*X(:,2)+b(9)*X(:,3)+b(10)*X(:,4)+b(11)*X(:,5)+b(12)*X(:,6)+...
%一次项

b(13)*X(:,1).*X(:,2)+b(14)*X(:,1).*X(:,3)+b(15)*X(:,1).*X(:,4)+b(16)*X(:,1).*X
(:,5)+... %交叉项

b(17)*X(:,1).*X(:,6)+b(18)*X(:,2).*X(:,3)+b(19)*X(:,2).*X(:,4)+b(20)*X(:,2).*X
(:,5)+...

b(21)*X(:,2).*X(:,6)+b(22)*X(:,3).*X(:,4)+b(23)*X(:,3).*X(:,5)+b(24)*X(:,3).*X
(:,6)+...
    b(25)*X(:,4).*X(:,5)+b(26)*X(:,4).*X(:,6)+b(27)*X(:,5).*X(:,6)+...

b(28)*sin(X(:,1))+b(29)*sin(X(:,2))+b(30)*sin(X(:,3))+b(31)*sin(X(:,4))+... %
三角项
    b(32)*sin(X(:,5))+b(33)*sin(X(:,6))+b(34);

tmp4=zeros(1,40); % 含对数拟合
model_4=@(b,X)b(1)*X(:,1).^2+b(2)*X(:,2).^2+b(3)*X(:,3).^2+b(4)*X(:,4).^2+b(5)
*X(:,5).^2+b(6)*X(:,6).^2+...%平方项

b(7)*X(:,1)+b(8)*X(:,2)+b(9)*X(:,3)+b(10)*X(:,4)+b(11)*X(:,5)+b(12)*X(:,6)+...
%一次项

b(13)*X(:,1).*X(:,2)+b(14)*X(:,1).*X(:,3)+b(15)*X(:,1).*X(:,4)+b(16)*X(:,1).*X
(:,5)+... %交叉项

b(17)*X(:,1).*X(:,6)+b(18)*X(:,2).*X(:,3)+b(19)*X(:,2).*X(:,4)+b(20)*X(:,2).*X
(:,5)+...

b(21)*X(:,2).*X(:,6)+b(22)*X(:,3).*X(:,4)+b(23)*X(:,3).*X(:,5)+b(24)*X(:,3).*X
(:,6)+...
    b(25)*X(:,4).*X(:,5)+b(26)*X(:,4).*X(:,6)+b(27)*X(:,5).*X(:,6)+...

b(28)*sin(X(:,1))+b(29)*sin(X(:,2))+b(30)*sin(X(:,3))+b(31)*sin(X(:,4))+... %
三角项
    b(32)*sin(X(:,5))+b(33)*sin(X(:,6))+...

b(34)*log(X(:,1))+b(35)*log(X(:,2))+b(36)*log(X(:,3))+b(37)*log(X(:,4))+...
```

```
        b(38)*log(X(:,5))+b(39)*log(X(:,6))+b(40); %对数项

tmp5=zeros(1,46); % 含指数拟合
model_5=@(b,X)b(1)*X(:,1).^2+b(2)*X(:,2).^2+b(3)*X(:,3).^2+b(4)*X(:,4).^2+b(5)
*X(:,5).^2+b(6)*X(:,6).^2+...%平方项

b(7)*X(:,1)+b(8)*X(:,2)+b(9)*X(:,3)+b(10)*X(:,4)+b(11)*X(:,5)+b(12)*X(:,6)+...
%一次项

b(13)*X(:,1).*X(:,2)+b(14)*X(:,1).*X(:,3)+b(15)*X(:,1).*X(:,4)+b(16)*X(:,1).*X
(:,5)+... %交叉项

b(17)*X(:,1).*X(:,6)+b(18)*X(:,2).*X(:,3)+b(19)*X(:,2).*X(:,4)+b(20)*X(:,2).*X
(:,5)+...

b(21)*X(:,2).*X(:,6)+b(22)*X(:,3).*X(:,4)+b(23)*X(:,3).*X(:,5)+b(24)*X(
:,3).*X(:,6)+...
    b(25)*X(:,4).*X(:,5)+b(26)*X(:,4).*X(:,6)+b(27)*X(:,5).*X(:,6)+...

b(28)*sin(X(:,1))+b(29)*sin(X(:,2))+b(30)*sin(X(:,3))+b(31)*sin(X(:,4))
+... %三角项
    b(32)*sin(X(:,5))+b(33)*sin(X(:,6))+...

b(34)*log(X(:,1))+b(35)*log(X(:,2))+b(36)*log(X(:,3))+b(37)*log(X(:,4))
+...%对数项
    b(38)*log(X(:,5))+b(39)*log(X(:,6))+...

b(40)*exp(X(:,1))+b(41)*exp(X(:,2))+b(42)*exp(X(:,3))+b(43)*exp(X(:,4))
+...%指数项
    b(44)*exp(X(:,5))+b(45)*exp(X(:,6))+b(46);

model={model_1,model_2,model_3,model_4,model_5};
tmp={tmp1,tmp2,tmp3,tmp4,tmp5};
p={};

e_list=zeros(3,sumM);
for flag=1:6
    tmp_list=zeros(269,sumM);
    for i=1:sumM
        [tmp_p,tmp_list(:,i)]= nlinfit(E,Y(:,flag),model{i},tmp{i}); %[系
数、残差]
        p{flag,i}=tmp_p;
    end
    e_list(flag,:)=mean(abs(tmp_list(:,:))./Y(:,flag)); % 相对误差
end

    save p p;save model model;
```

代码清单 2 力学性能预测模块

```
    % 本程序用于第二问的力学性能预测
clear;
load p;load TEST;load model; % 导入模型、参数、自变量数据
TEST;
result=zeros(7,7);  % 七组数据 每组预测 6 个参数
result(:,1)=TEST(:,1);
for i=1:6
    result(:,i+1)=model{5}(p{i,5},TEST(:,2:end)); % 预测第 i 个参数,用第五个
模型
end
disp("各个预测值")
disp(result(:,2:end))
xlswrite("第二问预测结果.xlsx",["熔炼号","屈服均值","抗拉均值",...
        "伸长率均值","屈服标准差","抗拉标准差","伸长率标准差";result]);
```

代码清单 3 绘制稳定性图像

```
import random
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.pyplot import MultipleLocator
import pandas as pd


df = pd.read_csv('yanshen_e4.csv')


fig = plt.figure()


plt.plot(df.iloc[:,3],df.iloc[:,0],linewidth=1,color="lightskyblue",la-
bel="predict")


plt.title('The Predictions of Extensibility',fontsize=18)
plt.tick_params(axis='both',which='major',labelsize=10)
plt.xlabel('E4',fontsize=15)
plt.ylabel('Extensibility',fontsize=15)


font = {'family':'Times New Roman','size': 15}
plt.legend(loc = 'upper left',prop=font)
plt.grid(linestyle='-')
    plt.savefig('./result1.jpg',dpi=500)
```

代码清单 4 整理出包含均值和标准差的数据

```
import pandas as pd
import csv
huaxue = pd.read_excel("new.xlsx")
lixue = pd.read_excel("LIXUE.xlsx")
hnrows = huaxue.shape[0]
hncols = huaxue.columns.size
lnrows = lixue.shape[0]
lncols = lixue.columns.size
i = 0
print(hncols)
j = 0
for i in range (hnrows):
    key = huaxue.iloc[i][0]
    if(huaxue.iloc[i][7] == 0):
        continue
    qufu = huaxue.iloc[i][7]
    kanla = huaxue.iloc[i][8]
    yanshen = huaxue.iloc[i][9]
    res1 = 0
    res2 = 0
    res3 = 0
    cnt = 0
    for j in range (lnrows):
        keynow = lixue.iloc[j][0]
        if(keynow == key):
            res1 += pow(lixue.iloc[j][2] - qufu , 2)
            res2 += pow(lixue.iloc[j][3] - kanla , 2)
            res3 += pow(lixue.iloc[j][4] - yanshen , 2)
            cnt += 1
    if(cnt != 0):
        res1 /= cnt - 1
        res2 /= cnt - 1
        res3 /= cnt - 1
        res1 = pow(res1 , 0.5)
        res2 = pow(res2, 0.5)
        res3 = pow(res3, 0.5)
```

```
        for j in range (lnrows):
            huaxue.at[i,"qufu_S"] = res1
            huaxue.at[i,"kanla_S"] = res2
            huaxue.at[i,"yanshen_S"] = res3
huaxue.to_excel("new1.xlsx",sheet_name="huaxue",index=False,encod-
ing="utf_8_sig")


plt.plot(df.iloc[:,3],df.iloc[:,0],linewidth=1,color="lightskyblue",la-
bel="predict")


plt.title('The Predictions of Extensibility',fontsize=18)
plt.tick_params(axis='both',which='major',labelsize=10)
plt.xlabel('E4',fontsize=15)
plt.ylabel('Extensibility',fontsize=15)


font = {'family':'Times New Roman','size': 15}
plt.legend(loc = 'upper left',prop=font)
plt.grid(linestyle='-')
    plt.savefig('./result1.jpg',dpi=500)
```

代码清单 5 生成测试稳定度的数据

```
    import pandas as pd
import csv
import random
huaxue = pd.read_excel("qufusensitivity.xlsx")
e1 = huaxue.iloc[0][1]
e2 = huaxue.iloc[0][2]
e3 = huaxue.iloc[0][3]
e4 = huaxue.iloc[0][4]
e6 = huaxue.iloc[0][6]
i = 1
j = 0
rate = 0
for i in range(11):
    rate=random.uniform(-0.05, 0.05)
    huaxue.at[i,"E2"] = (1 + rate) * e2
    huaxue.to_excel("qufusensitiv-
ity_data_e2.xlsx",sheet_name="huaxue",index=False,encoding="utf_8_sig")
```

代码清单 6 屈服性 GBDT 模型

```
    import numpy
import pandas
from spsspro.algorithm import supervised_learning
#生成案例数据
data_x = pandas.DataFrame({
    "A": numpy.random.random(size=100),
    "B": numpy.random.random(size=100)
})
data_y = pandas.Series(data=numpy.random.choice([1, 2], size=100),
name="C")
#GBDT 回归
result = supervised_learning.gbdt_regression(data_x=data_x,
data_y=data_y)
    print(result)
```

代码清单 7 融合抗拉性 GBDT 模型

```
import numpy
import pandas
from spsspro.algorithm import supervised_learning
#生成案例数据
data_x = pandas.DataFrame({
    "A": numpy.random.random(size=100),
    "B": numpy.random.random(size=100)
})
data_y = pandas.Series(data=numpy.random.choice([1, 2], size=100),
name="C")
#GBDT 回归
result = supervised_learning.gbdt_regression(data_x=data_x,
data_y=data_y)
    print(result)
```

代码清单 7 融合抗拉性 GBDT 模型

```
    import numpy
    import pandas
    from spsspro.algorithm import supervised_learning
```

```
    #生成案例数据
    data_x = pandas.DataFrame({
        "A": numpy.random.random(size=100),
        "B": numpy.random.random(size=100)
    })
    data_y = pandas.Series(data=numpy.random.choice([1, 2], size=100),
name="C")
    #GBDT 回归
    result = supervised_learning.gbdt_regression(data_x=data_x,
data_y=data_y)
    print(result)
```