中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生
数学建模竞赛

学　　校	贵州大学
参赛队号	20106570027
队员姓名	1.李滨
	2.唐华康
	3.柳波海

中国研究生创新实践系列大赛
“华为杯”第十七届中国研究生
数学建模竞赛

题 目 **降低汽油精制过程中的辛烷值损失模型**

摘 要：

催化裂化汽油的精制处理是我国绝大部分成品汽油脱硫保辛的关键步骤，使用硫含量过高的汽油会导致环境污染加剧，现有脱硫技术普遍会降低汽油中的辛烷值，辛烷值的降低会严重影响汽油的燃烧性能。本文针对汽油精制处理过程中操作变量之间高度强耦合和非线性问题，提出了利用关键操作变量预测辛烷值的机器学习模型，优化了实际化工过程需要的操作条件，分析了该装置对脱硫的性能，挖掘出了题中装置在减少辛烷值损失功能上的潜力，大大的提高了该产品的生态和经济效益。

针对问题一，对数据进行处理；① 对附件三前 40 行和后 40 行数据求平均值，分别代替附件一 285 号、313 号样本；② 对附件一中残缺数据较多、全部为空值的数据进行删除；③ 根据附件四中操作变量的取值范围，对附件一中不同时刻的操作变量求最大最小值，通过最大最小值初步比较范围，筛选出不在范围的操作变量，再具体到不在范围的数据，剔除与之对应的样本；④ 根据拉依达准则（3σ准则）去除非操作变量异常值。

针对问题二，为问题三中的建模选取主要变量，从而达到数据维度降低的目的，避免“维度灾难”，提高模型的精度和降低模型建立的难度。选择产品中的辛烷值作为选取问题三建模中的主要变量的关键因素。通过对产品中辛烷值与其他变量的皮尔逊相关性分析和对产品中硫含量与其他变量的皮尔逊相关性分析对 367 个变量进行初步筛选。最后，通过 BP 神经网络的预测模型进行综合分析，得到了 26 个主要变量。

针对问题三，通过分析问题和数据，我们得到的结论是应该预测的是辛烷值而非辛烷值损失。然后利用 XGBoost、LightGBM、随机森林、支持向量机等算法进行了模型建立和横向对比，最终发现 SVM 对于该预测任务误差最低、效果最好。最后我们通过利用实验的方法验证了“预测辛烷值比预测辛烷值损失更加合理“这个假设。

针对问题四，寻找操作变量的优化方案，实现辛烷值损失降幅最大化。以辛烷值损失降幅最大为优化目标，主要变量中 15 个操作变量为决策变量，产品含硫量不大于 5 μg/g 为其中一个约束条件，建立一个优化模型。通过粒子群算法、量子粒子群算法和差分量子粒子群优化算法对优化模型进行求解，从 301 个样本数据中筛选出辛烷值损失降幅大于等于 30%的样本。随着算法的改良，从 301 个样本数据中筛选出辛烷值损失降幅大于等于 30%的样本数量也在增长，其中差分量子粒子群算法的优化效果最为显著。

针对问题五，我们首先分析了第四个问题所得到的优化方案，然后列出了所有优化之后的操作变量，提出了一个操作变量优化方案调整策略，最后通过前面建立的模型验证了这个优化方案的可行性与可操作性。得到的结论是：该汽油精

制装置的“保辛烷”功能还没有完全开发出来，但是“脱硫”能力已经到达了极限。应用我们的操作变量优化方案能够最大限度的挖掘装置的“保辛烷脱硫”能力。

关键词： 3σ准则；维度灾难；皮尔逊相关性分析；XGBoost；LightGBM；随机森林；支持向量机；粒子群算法；量子粒子群算法；差分量子粒子群优化算法；“保辛烷脱硫”能力

目录

第一章 前言

1.1 研究背景

当代我国汽车工业得到了迅速的发展，汽油作为汽车主要燃料，它的消耗量也大大增加，汽车尾气所造成的环境污染也日益加剧。响应国家“既要金山银山，又要绿水青山”的号召，加强对汽车尾气污染的治理，国家最新出台的国六汽油燃料标准中有着更加严格的要求[1]。其中，要求硫含量不高于$10ppm$,烯烃含量不高于$18wt\%$。对比国五汽油标准，除了对硫含量有更加严格的要求外，烯烃含量也有更加严格的限制[2]。

催化裂化汽油也称 FCC 汽油，超过 70%的汽油是由催化裂化生产得到，商品汽油中大多数烯烃和近 85%以上硫化物的主要来源就是催化裂化汽油，所以催化裂化汽油的脱硫降烯是我国清洁汽油生产的关键一步。选择性加氢脱硫技术可以有效降低汽油中的硫含量，但是也会造成很大的辛烷值损失[3]。辛烷值是评价汽油性能的重要参考依据，并且辛烷值损失会造成巨大的经济效益损失。

面对当前限烯的燃油标准，在降低环境污染，保证汽油产品脱硫效果的前提下，降低辛烷值损失成为当前制约 FCC 汽油生产的一个难点[4]。

1.2 问题重述

依据从催化裂化汽油精制装置采集的 325 个数据样本（每个数据样本都有 354 个操作变量），通过数据挖掘技术来建立汽油辛烷值（RON）损失的预测模型，并给出每个样本的优化操作条件，在保证汽油产品脱硫效果（本次建模要求产品硫含量不大于 5μg/g）的前提下，尽量降低汽油辛烷值损失在 30%以上。

问题 1：数据处理：参考近 4 年的工业数据的预处理结果，依“样本确定方法”（附件二）对 285 号和 313 号数据样本进行预处理，并将处理后的数据分别加入到附件一中相应的样本号中，供下面研究使用。

问题 2：寻找建模主要变量， 建立降低辛烷值损失模型涉及包括 7 个原料性质、2 个待生吸附剂性质、2 个再生吸附剂性质、2 个产品性质等变量以及另外 354 个操作变量（共计 367 个变量），工程技术应用中经常使用先降维后建模的方法，这有利于忽略次要因素，发现并分析影响模型的主要变量与因素。根据提供的 325 个样本数据（见附件一），通过降维的方法从 367 个操作变量中筛选出 30 个以下以下的建模主要变量，使之尽可能具有代表性、独立性，并详细说明建模主要变量的筛选过程及其合理性。（提示：请考虑将原料的辛烷值作为建模变量之一）。

问题 3：建立辛烷值（RON）损失预测模型：采用上述样本和建模主要变量，通过数据挖掘技术建立辛烷值（RON）损失预测模型，并进行模型验证。

问题 4：主要变量操作方案的优化，要求在保证产品硫含量不大于 5μg/g 的前提下，利用我们的模型获得 325 个数据样本，找出辛烷值（RON）损失降幅大于 30%的样本对应的主要变量优化后的操作条件（优化过程中原料、待生吸附剂、再生吸附剂的性质保持不变，以它们在样本中的数据为准）。

问题 5：模型的可视化展示，对 133 号样本（原料性质、待生吸附剂和再生吸附剂的性质数据保持不变，以样本中的数据为准），以图形展示其主要操作变量优化调整过程中对应的汽油辛烷值和硫含量的变化轨迹。

1.3 评价指标

为了更清楚地展示模型预测结果，本文选取了均方根误差（RMSE）、平均绝对误差（MAE）、平均绝对百分比误差（MAPE）三种评价指标对产品的辛烷值预测模型进行评价，再根据预测的产品辛烷值得到预测的 RON 损失值。下面是三种评价指标的详细介绍。

1.3.1 均方根误差

它是观测值与真值偏差的平方和观测次数 n 比值的平方根，当对某一量进行甚多次的测量时，取这一测量列真误差的均方根差(真误差平方的算术平均值再开方)，称为标准偏差，以 σ 表示。σ 反映了测量数据偏离真实值的程度，σ 越小，表示测量精度越高，因此可用 σ 作为评定这一测量过程精度的标准。在实际测量中，观测次数 n 总是有限的，真值只能用最可信赖（最佳）值来代替。方根误差对一组测量中的特大或特小误差反映非常敏感，所以，均方根误差能够很好地反映出测量的精密度。

$$RMSE(X,h)=\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(h(x_i)-y_i\right)^2} \tag{1.1}$$

其中，$h(x_i)$ 为预测值，y_i 为真实值。

1.3.2 平均绝对误差

对同一物理量进行多次测量时，各次测量值及其绝对误差不会相同，我们将各次测量的绝对误差取绝对值后再求平均值，并称其为平均绝对误差，即：$\Delta=\left(|\Delta 1|+|\Delta 2|+\ldots|\Delta n|\right)/n$（Δ为平均绝对误差；Δ1、Δ2、…Δ*n* 为各次测量的绝对误差）。

$$MAE=\frac{1}{m}\sum_{i=1}^{m}|h(x_i)-y_i| \tag{1.2}$$

平均绝对误差与平均误差相比，平均绝对误差由于离差被绝对值化，不会出现正负相抵消的情况，因而，平均绝对误差能更好地反映预测值误差的实际情况。

1.3.3 平均绝对百分比误差

将预测值与真实值的相对误差进行求和，然后在除以观测次数，然后取百分数的过程就是平均绝对百分比误差的求取过程。平均绝对百分比误差之所以可以描述准确度，是因为平均绝对百分比误差本身就是用于衡量预测准确性的统计指标，它的取值范围为[0,+∞)，MAPE 为 0%表示完美模型，MAPE 大于 100 %则表示劣质模型。

$$MAPE=\frac{100\%}{n}\sum_{i=1}^{m}\left|\frac{h(x_i)-y_i}{y_i}\right| \tag{1.3}$$

1.4 本文的架构设计

文章的结构如下：

第一部分：前言。主要对汽油精制过程中辛烷值损失的相关历史背景进行了描述，引出了汽油精制过程中保护环境与保证经济效益的矛盾，初步分析在保证

汽油产品脱硫效果的前提下降低辛烷值损失的必要性，对问题进行了重述与简要分析。

第二部分：问题 1 解答，对数据进行处理。① 对附件三前 40 行和后 40 行数据求平均值，分别代替附件一 285 号、313 号样本；② 对附件一中残缺数据较多、全部为空值的变量进行删除；③ 根据附件四中操作变量的取值范围，对附件一中不同时刻的操作变量求最大最小值，通过最大最小值初步比较范围，筛选出不在范围的操作变量，再具体到不在范围的数据，剔除与之对应的样本；④ 根据拉依达准则（3σ准则）去除非操作变量异常值。

第三部分：问题 2 解答，寻找建模主要变量。为问题三中的建模选取主要变量，从而达到数据维度降低的目的，避免“维度灾难”，提高模型的精度和降低模型建立的难度。选择产品中的辛烷值作为选取问题三建模中的主要变量的关键因素。通过对产品中辛烷值与其他变量的皮尔逊相关性分析和对产品中硫含量与其他变量的皮尔逊相关性分析对 367 个变量进行初步筛选。最后，通过 BP 神经网络的预测模型进行综合分析，得到了 26 个主要变量。

第四部分：问题 3 的解答，建立辛烷值（RON）损失预测模型。通过分析问题和数据，我们得到的结论是应该预测的是辛烷值而非辛烷值损失。然后利用 XGBoost、LightGBM、随机森林、支持向量机等算法进行了模型建立和横向对比，最终发现 SVM 对于该预测任务误差最低、效果最好。最后我们通过利用实验的方法验证了“预测辛烷值比预测辛烷值损失更加合理”这个假设。

第五部分：问题 4 的解答，主要变量操作方案的优化。寻找操作变量的优化方案，实现辛烷值损失降幅最大化。以辛烷值损失降幅最大为优化目标，主要变量中 15 个操作变量为决策变量，产品含硫量不大于 5μg/g 为其中一个约束条件，建立一个优化模型。通过粒子群算法、量子粒子群算法和差分量子粒子群优化算法对优化模型进行求解，从 301 个样本数据中筛选出辛烷值损失降幅大于等于 30%的样本。随着算法的改良，从 301 个样本数据中筛选出辛烷值损失降幅大于等于 30%的样本数量也在增长，其中差分量子粒子群算法的优化效果最为显著。

第六部分：问题 5 的解答，模型的可视化展示。我们首先分析了第四个问题所得到的优化方案，然后列出了所有优化之后的操作变量，提出了一个操作变量优化方案调整策略，最后通过前面建立的模型验证了这个优化方案的可行性与可操作性。得到的结论是：该汽油精制装置的“保辛烷”功能还没有完全开发出来，但是“脱硫”能力已经到达了极限。应用我们的操作变量优化方案能够最大限度的挖掘装置的“保辛烷脱硫”能力。

第七部分：模型评价。对模型的优缺点进行总结评价。

本文的研究框架如图 1-1 所示。

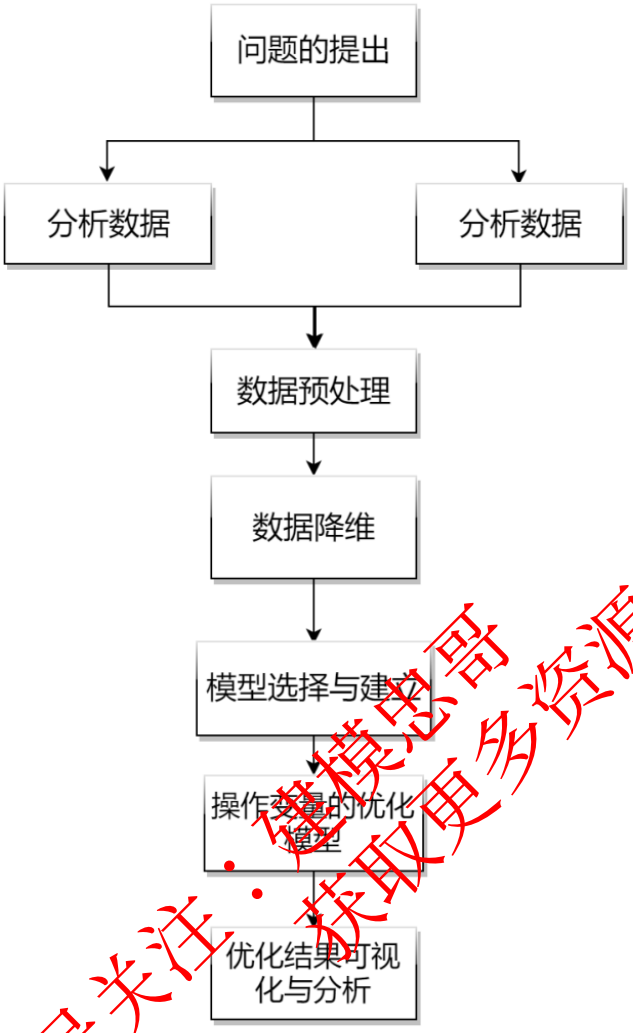


图 1-1 研究框架图

第二章 数据处理

2.1 问题分析

第一部分要求对数据进行处理，这是其他四个问题的基础，第一步先按照问题要求替换 285 号和 313 号样本；第二步再去除异常值较多的变量；第三步按照操作变量取值范围进行筛选；第四步依照 3σ 准则对非操作变量异常值进行分析，剔除一些不合理的样本，完成对数据的处理，保留优质数据。

2.2 拉依达准则（3σ 准则）介绍

拉依达准则又称为 3σ 准则，先假设一组数据只有随机误差，对它进行计算处理得到标准偏差，再按一定的概率确定一个范围，凡是超过了这个范围的误差，就不属于随机误差，含有该误差的数据就应该被删除掉[5]。

在正态分布里面，σ 表示的标准差，μ 表示均值，$x=\mu$ 是图像的对称轴。3σ 准则为：

数值分布在 $(\mu-\sigma,\mu+\sigma)$ 的概率是 0.6826；

数值分布在 $(\mu-2\sigma,\mu+2\sigma)$ 的概率是 0.9544；

数值分布在 $(\mu-3\sigma,\mu+3\sigma)$ 的概率是 0.9974。

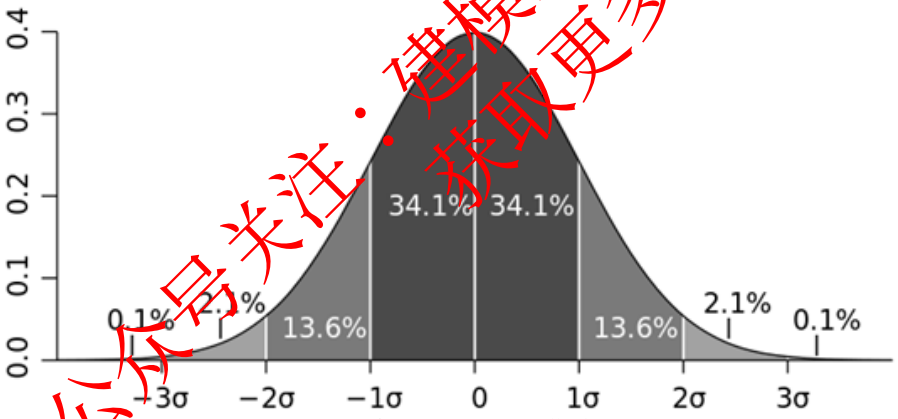


图 2-1 标准正态分布曲线

可以看出，Y 的取值基本上全部集中在 $(\mu-3\sigma,\mu+3\sigma)$ 范围内，超出这个区间的可能性不大于 0.3%。

拉依达准则是建立在正态分布的等精度重复测量基础上而造成奇异数据的干扰，或噪声难以满足正态分布。如果一组测量数据中某个测量值的残余误差的绝对值 $vi>3\sigma$，则该测量值为坏值，应剔除。通常把等于 $\pm3\sigma$ 的误差作为极限误差，对于正态分布的随机误差，落在 $\pm3\sigma$ 以外的概率只有 0.27%，它在有限次测量中发生的可能性很小，故存在 3σ 准则。3σ 准则是最常用也是最简单的粗大误差判别准则，它一般应用于测量次数充分多 $(n\geq30)$ 或当 $n>30$ 做粗略判别时的情况。

2.3 数据处理

2.3.1 附件三数据处理

（1）观察附件三数据，发现附件三中数据存在缺失值、异常值，取其前后

两个小时数据的平均值进行代替；

（2）对附件三前 40 行和后 40 行数据求平均值，分别代替附件一 285 号、313 号数据；

2.3.2 附件一数据处理

（1）找出附件一中残缺数据较多、全部为空值的数据进行删除，同时剔除异常值较多的操作变量，比如精制汽油出装置硫含量、原料缓冲罐液位、再生烟气氧含量；

（2）求出附件一中不同时刻的操作变量求最大最小值，对比附件四中操作变量的取值范围，对超过取值范围的操作变量进行排序，筛选出不在范围内的数据，删除对应样本。我们通过对剩下的操作变量进行上下限处理，我们发现只有变量 S-ZORB.FT_1202.TOTAL 的取值稍有异常，图 2-2 是具体的异常情况。

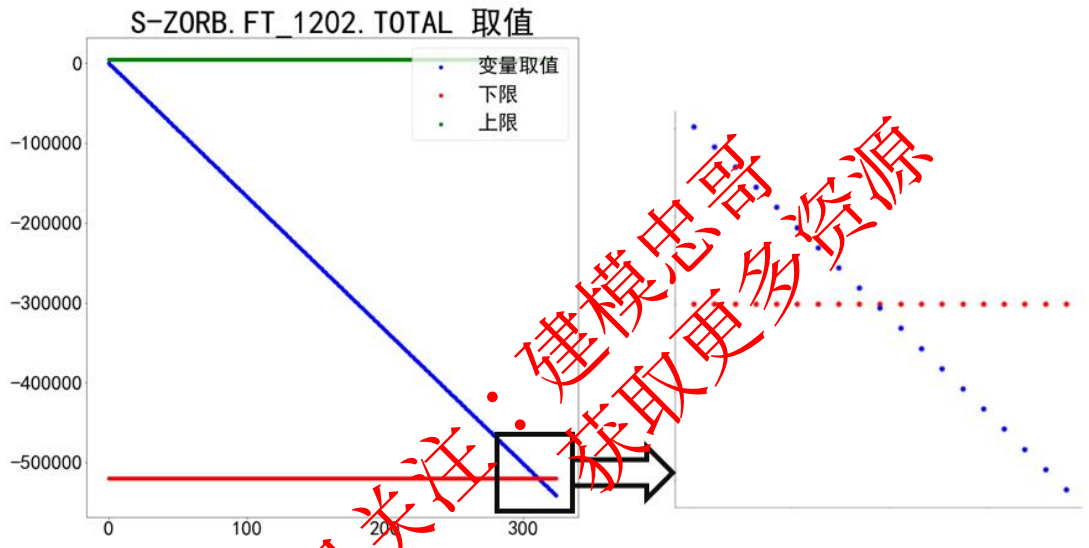


图 2-2 变量 S-ZORB.FT_1202.TOTAL 取值情况

利用上下限幅值原则进行分析，发现共有 10 个样本需要剔除，如表 2-1，它们对应的样本编号为：

表 2-1 不在取值范围内的样本编号

样本编号	316	317	318	319	320	321	322	323	324	325

（3）根据拉依达准则（3σ 准则）去除非操作变量异常值。分别求出 14 个非操作变量的均值 μ、标准差 σ，以及对应区间 $(\mu-3\sigma,\mu+3\sigma)$，如表 2-2、2-3、2-4 所示：

表 2-2 原料性质中变量的 μ、σ，以及区间 $(\mu-3\sigma,\mu+3\sigma)$

	原料性质						
	硫含量	辛烷值	饱和烃	烯烃	芳烃	溴值	密度
均值 μ	221.98	89.67	53.41	24.54	22.04	54.11	727.02
标准差 σ	64.82	0.97	4.56	4.93	1.86	7.15	4.27
$\mu-3\sigma$	27.53	86.77	39.73	9.77	16.45	32.66	714.22
$\mu+3\sigma$	416.42	92.56	67.10	39.32	27.64	75.56	739.83

表 2-3 产品性质中变量的 μ 、 σ ，以及区间 $(\mu-3\sigma,\mu+3\sigma)$

	产品性质		
	焦炭	S	RON 损失
均值 μ	3.87	88.41	1.25
标准差 σ	1.49	1	0.24
$\mu-3\sigma$	-0.59	85.40	0.55
$\mu+3\sigma$	8.32	91.41	1.96

表 2-4 待生、再生吸附剂性质中变量的 μ 、 σ ，以及区间 $(\mu-3\sigma,\mu+3\sigma)$

	待生吸附剂性质		再生吸附剂性质	
	焦炭	S	焦炭	S
均值 μ	2.81	7.8	1.4	5.72
标准差 σ	1.74	2.07	1.05	1.73
$\mu-3\sigma$	-2.41	1.57	-1.76	0.55
$\mu+3\sigma$	8.03	14.02	4.56	10.90

根据各个非操作变量的 $(\mu-3\sigma,\mu+3\sigma)$ 区间，找出不在区间内的数据，再删除对应样本，发现共有 14 个样本需要剔除，它们对应的样本编号为：

表 2-5 非操作变量不在 $(\mu-3\sigma,\mu+3\sigma)$ 区间对应的样本编号

样本编号	13	90	91	92	93	141	193
	203	205	210	212	286	295	314

2.4 总结

数据处理按照问题要求进行、替换数据、去除无效数据、剔除超过取值范围数据、删除不满足 3σ 准则数据。处理后的优质数据还剩 301 个样本，345 个变量，在第三问中也能很好预测 RON 损失，证明处理数据有效。

第三章　寻找建模主要变量

问题二要求为问题三中的建模选取主要变量，从而达到数据维度降低的目的，避免“维度灾难”，提高模型的精度和降低模型建立的难度。选择产品中的辛烷值作为选取问题三建模中的主要变量的关键因素。通过对产品中辛烷值与其他变量的皮尔逊相关性分析和对产品中硫含量与其他变量的皮尔逊相关性分析对 367 个变量进行初步筛选。最后，通过 BP 神经网络的预测模型进行综合分析。

3.1 问题分析

与辛烷值损失相关的因素包括 7 个原料性质、2 个产品性质、2 个待生吸附剂性质、2 个再生吸附剂性质和 354 个操作变量总共 367 个相关因素。若直接采用这 367 个相关因素对辛烷值损失建立预测模型，容易造成很多问题。

首先是特征变量高达 367 个，数据维度太高，增加模型建立的难度，使得学习算法会具有较高的时间和空间复杂度。其次，随着数据维度的增加，模型的计算量会呈指数增长，从而造成“维度灾难”[6]。最后，高维数据也降低了学习算法的精度和泛化能力，并且容易造成过拟合现象，从而导致模型的精度大幅度下降。

因此，为了有利于问题三中辛烷值损失预测模型的建立，在问题二中我们需要在 367 个相关因素中，筛选出与辛烷值损失高度相关的关键因素，忽略与辛烷值损弱相关的次要因素，从而达到对数据进行降维的目的，尽可能地降低问题三中辛烷值损失预测模型的建模难度，避免维度灾难和提高辛烷值损失预测模型的精度。

由于辛烷值损失是原料中的辛烷值与产品中的辛烷值之差，无法很好的表征其与产品之间的属性联系。若直接用辛烷值损失与 367 个变量进行相关性分析，容易造成很多关键因素的丢失，从而造成过度降维。然而，原料中的辛烷值是原料的固有属性，产品中的辛烷值是经过工艺操作后产品的性质。产品中的辛烷值是催化裂化汽油精制过程的结果，即它是 354 个操作变量的因果体现。并且辛烷值的损失值可以由原料中的辛烷值与产品中的辛烷值之差来获得，故采用产品辛烷值来进行相关性分析和预测比辛烷值损失来得更加合理并且可靠。

在问题四对主要操作变量的优化中，降低产品中的硫含量可以增强产品的环保性，因此需要对硫含量有不大于 5μg/g 的限制。同时，在问题五中需要将主要操作变量优化调整过程中对应的汽油硫含量的变化轨迹可视化，所以在后续问题的研究中，产品中硫含量对操作变量变化的响应也是需要研究的。这就要求我们对产品中的硫含量建立一个预测模型。同理，我们要从 367 个变量中筛选出与产品中含硫量高度相关的主要因素，忽略次要因素。

综上所述，为了更加充分体现关键变量选取的合理性和可靠性，在这里我们联合产品中辛烷值和含硫量的相关性分析，科学地选取关键变量，降低数据的维度。

由于操作变量之间具有高度非线性和相互强耦合的关系，因此在此次数据降维中采用皮尔逊相关性分析。依次对产品中辛烷值与剩余的 365 个变量进行皮尔逊相关性分析；同理，也是对产品中硫含量与剩余的 365 个变量进行皮尔逊相关性分析。然后联合两者的分析结果，初步筛选出关键变量。

3.2 皮尔逊相关性分析

皮尔逊相关系数，又称为皮尔逊积矩相关系数[7]，是用于度量两个变量 X 和

Y之间的相关性，其值介于-1与1之间。一般用于分析两个连续变量之间的关系，是一种线性相关系数，公式为：

$$r=\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}} \tag{3.1}$$

其中，r是两个变量X和Y之间的皮尔逊相关系数，$x_i(i=1,2,\cdots,n)$是向量X的元素，$y_i(i=1,2,\cdots,n)$是向量Y的元素。

相关系数r的取值范围为$-1\le r\le 1$。

$$\begin{cases} r>0，\text{为正相关，}r<0\text{为负相关} \\ |r|=0，\text{表示不存在线性相关} \\ |r|=1，\text{表示完全线性相关} \end{cases}$$

当$0<|r|<1$时，r表示两个变量之间存在不同程度的线性相关：

① $0.8<|r|<1$，表示两个变量是极强相关；

② $0.6<|r|<0.8$，表示两个变量是强相关；

③ $0.4<|r|<0.6$，表示两个变量是中等程度相关；

④ $0.2<|r|<0.4$，表示两个变量是弱相关；

⑤ $0<|r|<0.2$，表示两个变量是极弱相关或无相关。

当两个变量的标准差都不为零时，相关系数才有意义，皮尔逊相关系数适用于：

(1) 两个变量之间是线性关系，都是连续数据。

(2) 两个变量的总体是正态分布，或接近正态的单峰分布。

(3) 两个变量的观测值是成对的，每对观测值之间相互独立。

3.3 与产品中辛烷值相关皮尔逊相关性分析

3.3.1 产品中辛烷值与其他变量的皮尔逊相关性分析

在这一步中，本研究使用了 IBM SPSS Statistics 26.0 对产品中的辛烷值和其他变量逐次进行了皮尔逊相关性分析。将计算得到的 365 个皮尔逊系数导入 Excel 2016 中，结合 Excel 2016 和 IBM SPSS Statistics 26.0 对得到的数据进行分析。

如图 3-1 所示，本次研究选取了部分的皮尔逊系数进行绘制了产品中辛烷值与其他变量的皮尔逊相关系数的变化趋势图，其中包含了所有皮尔逊相关系数绝对值大于 0.4 的情况。表 3-1 是产品中辛烷值与其他变量的皮尔逊相关系数绝对值大于 0.4 的所有情况。

结合表 3-1 可知，图 3-1 中皮尔逊相关系数$r>0.8$的点是产品中的辛烷值与原材料中辛烷值的皮尔逊相关系数，即他们属于极强相关。因为产品中的辛烷值是原材料中辛烷值由于工业流程造成了损失，从而形成了产品中的辛烷值。所以两者会存在极强的相关性。

其他的皮尔逊相关系数主要集中在$-0.4<r<0.4$之间，这些变量与产品中的辛烷值是弱相关、极弱相关或者无关。而分布在$0.4<|r|<0.6$之间的皮尔逊相关系数则很少，具体如表 3-1 所示。从表中可知，除了原材料的辛烷值外，原材料

的硫含量、饱和烃和烯烃与产品中的辛烷值呈现中等相关。剩余 10 个与产品中的辛烷值呈现中等相关的是操作变量。从饼图 3-2 中可知，皮尔逊相关系数 $|r|>0.4$ 的情况在总的相关变量中只占了很少的一部分，大约是 3.8%。

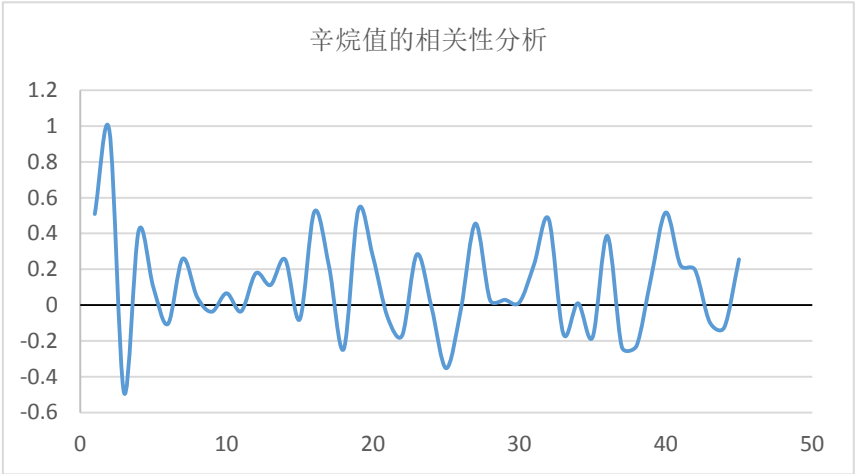


图 3-1 产品中辛烷值与其他变量的皮尔逊相关系数的部分变化趋势

原料中烯烃含量高会使烯烃加氢饱和反应更剧烈，辛烷值损失增大，低碳数的烯烃相对于高碳数的烯烃更易加氢饱和，而汽油组分中 C5、C6 烯烃在辛烷值占据的比例最高，则加氢饱和后辛烷值损失会更大。因此，本研究中将原料性质中的硫含量、辛烷值、饱和烃和烯烃作为问题三建模中的主要变量的一部分。

下面将会对原材料的辛烷值与操作变量的相关性进行详细分析。

表 3-1 产品中辛烷值与其他变量的皮尔逊相关系数绝对值大于 0.4

变量名称	皮尔逊相关系数	变量名称	皮尔逊相关系数
硫含量(原料性质)	0.5087	辛烷值(原料性质)	0.9733
饱和烃(原料性质)	-0.4899	烯烃(原料性质)	0.4158
S-ZORB.FT_1001.PV	0.5219	S-ZORB.TE_1001.PV	0.5327
S-ZORB.TE_1105.PV	0.4553	S-ZORB.TE_1201.PV	0.4804
S-ZORB.FT_5201.PV	0.5172	S-ZORB.TE_1101.DACA	0.4693
S-ZORB.TE_5002.DACA	0.4819	S-ZORB.TE_7108B.DACA	-0.4047
S-ZORB.PT_7107B.DACA	-0.4502	S-ZORB.PT_7103B.DACA	-0.4433

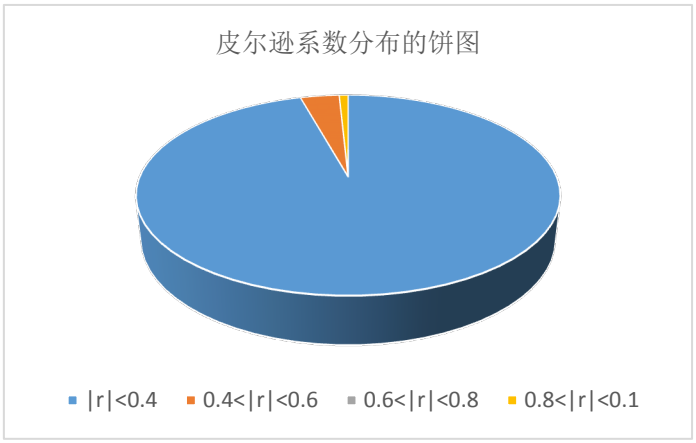


图 3-2 皮尔逊相关系数各部分的占比

3.3.2 产品中辛烷值与温度有关的操作变量之间相关性分析

在与产品中辛烷值呈中等程度相关的操作变量中，S-ZORB.TE_1001.PV、S-ZORB.TE_1105.PV 、 S-ZORB.TE_1201.PV 、 S-ZORB.TE_1101.DACA 、S-ZORB.TE_5002.DACA 和 S-ZORB.TE_7108B.DACA 分别是原料进装置温度、原料换热器管程总管进口温度、D104 温度、E-101 壳程出口总管温度、C-201 下部进料管温度和 K-101B 右排气温度。

反应温度是影响产品中辛烷值的一个很重要的因素，因为烯烃加氢饱和反应是一个强放热反应，所以提高反应温度可以抑制烯烃饱和反应。在其它条件不变的情况下随着反应温度的升高，产品的辛烷值损失逐渐减小，这也说明高温可以抑制烯烃饱和反应。所以反应温度对辛烷值损失有着重要的影响。特别低，这些点位的温度操作更是与产品的辛烷值有着中等程度的相关性。

所以本次研究中将 S-ZORB.TE_1001.PV 、 S-ZORB.TE_1105.PV 、S-ZORB.TE_1201.PV 、S-ZORB.TE_1101.DACA 、S-ZORB.TE_5002.DACA 和 S-ZORB.TE_7108B.DACA 这 6 个操作变量作为问题三建模中的主要变量的一部分。

3.3.3 产品中辛烷值与流量有关的操作变量之间相关性分析

在与产品中辛烷值呈中等程度相关的操作变量中，S-ZORB.FT_1001.PV 和 S-ZORB.FT_5201.PV 是催化汽油进装置总流量和产品汽油出装置流量。

汽油进出口装置的流量变化会影响原料在 S-ZORB 装置中的完全反应程度，流量越大，原料未充分反应的程度越大，从而产品中辛烷值的损失越小。因此这些点位的流量操作对产品的辛烷值有着重要的影响。

所以本次研究中将 S-ZORB.FT_1001.PV 和 S-ZORB.FT_5201.PV 这 2 个操作变量作为问题三建模中的主要变量的一部分。

3.3.4 产品中辛烷值与压力有关的操作变量之间相关性分析

在与产品中辛烷值呈中等程度相关的操作变量中，S-ZORB.PT_7107B.DACA 和 S-ZORB.PT_7103B.DACA 是 K-101B 排气压力和 K-101B 进气压力。

烯烃加氢反应是体积减小的化合反应，从反应动力学原理得知，降低反应压力、氢分压都有利于降低产品辛烷值损失，所以反应压力对产品中的辛烷值有着深远的影响。特别地，K-101B 点位对压力的操作，更是会影响反应的变化。因此 K-101B 排气压力和 K-101B 进气压力与产品中辛烷值呈中等程度的相关性。

所以本次研究中将 S-ZORB.PT_7107B.DACA 和 S-ZORB.PT_7103B.DACA 这 2 个操作变量作为问题三建模中的主要变量的一部分。

3.4 产品中硫含量与其他变量的皮尔逊相关性分析

在这一步中，与对产品中辛烷值与其他变量的皮尔逊相关性分析使用的方法相同。本研究中仍然使用了 IBM SPSS Statistics 26.0 对产品中的辛烷值和其他变量逐次进行了皮尔逊相关性分析。将计算得到的 365 个皮尔逊系数导入 Excel 2016 中，结合 Excel 2016 和 IBM SPSS Statistics 26.0 对得到的数据进行分析。

图 3-3 是产品中硫含量与其他变量的皮尔逊相关系数的部分变化趋势图，其中包含了所有皮尔逊相关系数绝对值大于 0.4 的情况。表 3-2 是产品中硫含量与其他变量的皮尔逊相关系数绝对值大于 0.4 的所有情况。

产品中硫含量与其他变量皮尔逊相关系数主要集中在 $-0.4 < r < 0.4$ 之间，这些变量与产品中的硫含量是弱相关、极弱相关或者无关。分布在 $0.4 < |r| < 0.6$ 之间的皮尔逊相关系数很少，具体如表 3-2 所示。在与产品中硫含量呈中等程度相关的操作变量 S-ZORB.FC_2801.PV、S-ZORB.TC_2801.PV、S-ZORB.FT_3304.DACA、S-ZORB.PDT_2605.DACA 和 S-ZORB.PC_1001A.PV 分别是还原器流化氢气流量、还原器温度、无名节点、D-123 凝结水入口流量和 D101 原料缓冲罐压力。

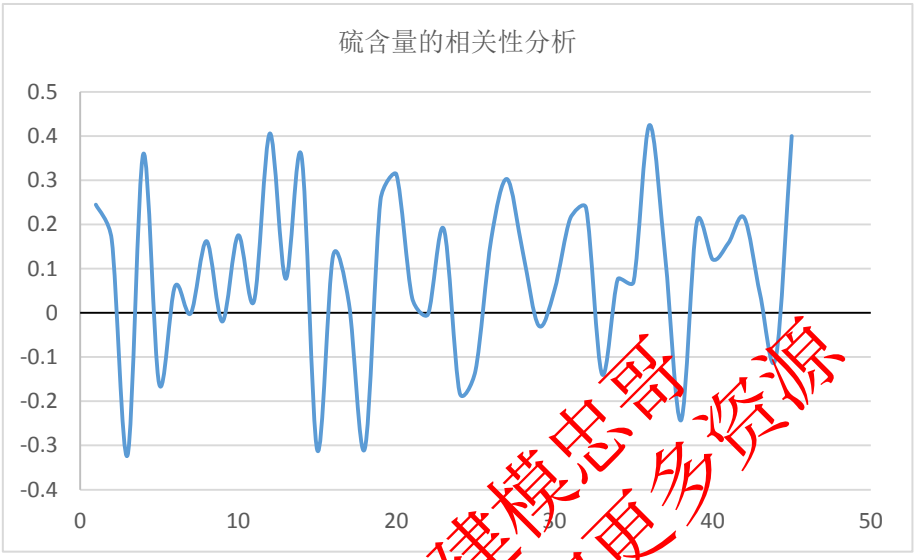


图 3-3 产品中硫含量与其他变量的皮尔逊相关系数的部分变化趋势

还原器流化氢气流量、还原器温度、D-123 凝结水入口流量和 D101 原料缓冲罐压力，这些变量涉及了流量、温度和压力。其中温度和压力会影响还原反应的效率和充分程度，而流量则会响应原料在装置中的反应时间和程度。这些因素都与产品中的含硫量息息相关，尤其是这些点位的操作变量，与产品中硫含量呈现中等程度的相关性。

表 3-2 产品中硫含量与其他变量的皮尔逊相关系数绝对值大于 0.4

变量名称	皮尔逊相关系数
S-ZORB.FC_2801.PV	0.4059
S-ZORB.TC_2801.PV	0.4248
S-ZORB.FT_1204.TOTAL	0.4072
S-ZORB.FT_3304.DACA	-0.4040
S-ZORB.PC_1001A.PV	-0.4020

从饼图 3-4 中可知，皮尔逊相关系数 的情况在总的相关变量中只占了很少的一部分，大约是 1.4%。

为了在问题三的模型中，体现对产品硫含量的影响，本研究选取 S-ZORB.FC_2801.PV 、 S-ZORB.TC_2801.PV 、 S-ZORB.FT_3304.DACA 、 S-ZORB.PDT_2605.DACA 和 S-ZORB.PC_1001A.PV 这 5 个操作变量作为问题三建模中的主要变量的一部分。

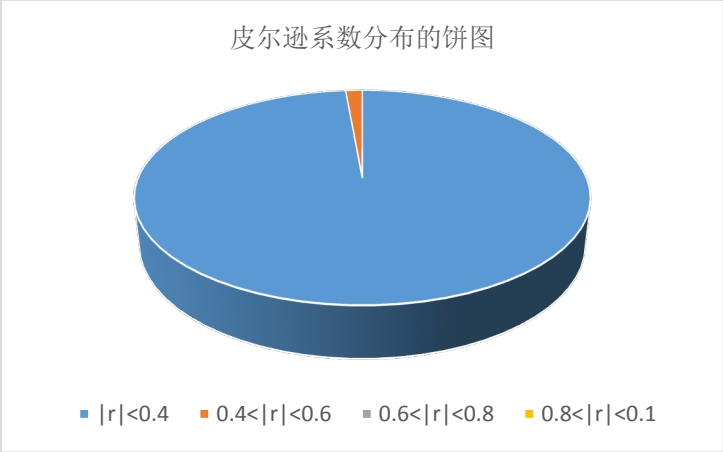


图 3-4 皮尔逊相关系数各部分的占比

3.5 综合分析

S-Zorb 装置催化汽油包含了进料与脱硫反应、吸附剂再生、吸附剂循环和产品稳定四个部分。该技术基于吸附作用原理对汽油进行脱硫，通过吸附剂选择性地吸附含硫化合物中的硫原子而达到脱硫目的。由 S-Zorb 装置工作过程可知，待生吸附剂性质和再生吸附剂性质是与产品中辛烷值和硫含量有着重大的关系，它们影响着辛烷值的损失大小和产品脱硫量的大小。

同时原料性质中的芳烃、溴值和密度在化学反应原理中，对产品中辛烷值和硫含量也密切相关。但是这 7 个变量在各产品中辛烷值的皮尔逊相关性分析中，呈现弱相关或者极弱相关。而这有可能历史数据的不完备使得无法通过皮尔逊相关性分析呈现出与产品中辛烷值和硫含量的强相关性。为了验证这 7 个变量是否与产品中辛烷值有强相关性，在此进行了一个算法案例验证。

3.5.1 算法案例研究

为了探究这 7 个变量，包括原料性质中的芳烃、溴值和密度与两个待生吸附剂性质和两个再生吸附剂性质，它们是否与产品中辛烷值有强相关性，本研究在次做了一个对照实验。采用问题三中建立好的一个 BP 神经网络对产品中的辛烷值进行预测（该 BP 神经网络模型会在下个问题中进行详细的阐述），实验一采用的数据样本是包含了这 7 个变量在内和已经选取好的 19 个变量，即总共具有 26 个主要变量的数据样本。实验二采用的数据样本是仅仅包含已经选取好的 19 个主要变量的数据样本。其余设置在这两次实验中均是相同的。

图 3-5 是实验一中对产品中辛烷值的预测结果，图 3-6 是实验二中对产品中辛烷值的预测结果，表 3-3 是两次实验中预测结果的均方根误差 RMSE 的值。由两幅图可知，实验一中对产品中辛烷值的预测曲线的拟合准确度是高于实验二的。由表 3-3 中可知，实验一的预测结果的 RMSE 值小于实验二的预测结果的 RMSE 值，这表明实验一的预测精度要高于实验二。而这两者之间的区别则是由样本数据中是否包含了上述的那 7 个变量造成的。

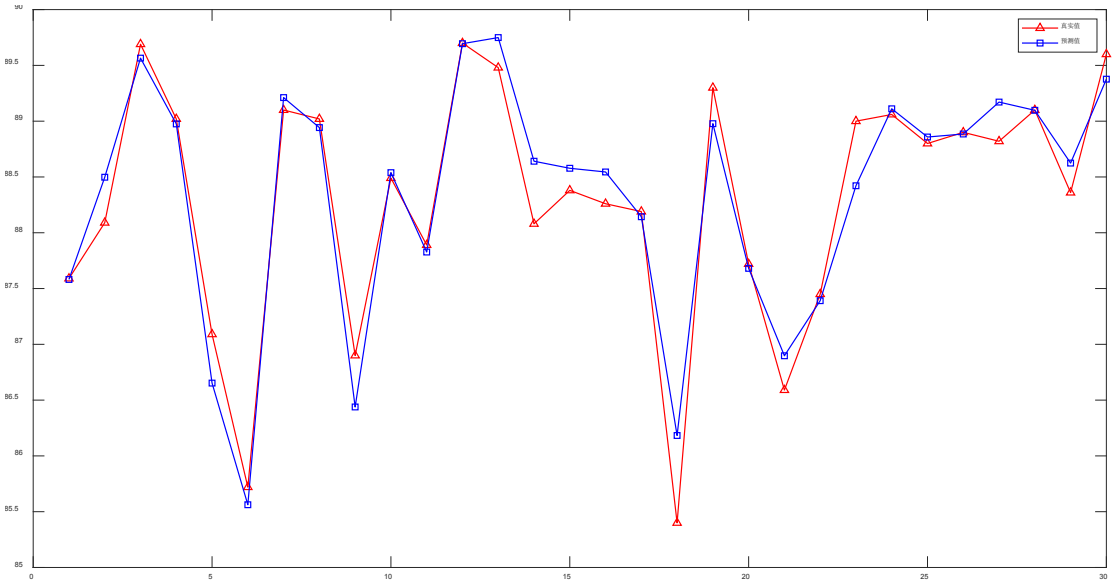


图 3-5 实验一对产品中辛烷值的预测结果

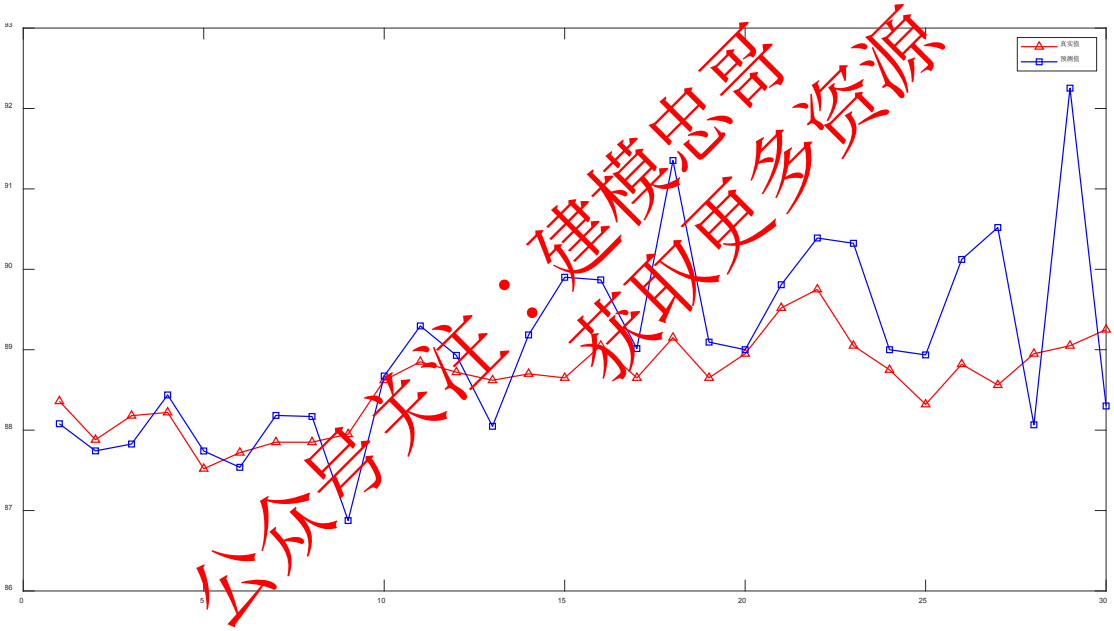


图 3-6 实验二对产品中辛烷值的预测结果

表 3-3 预测结果的 RMSE

实验序号	预测结果的 RMSE
实验一	0.1096
实验二	0.3758

3.5.2 算法案例分析

实验一中预测曲线拟合度的准确性高于实验二中预测曲线拟合度的准确性，是因为实验二中的数据样本中不包含这 7 个变量，造成了预测模型的欠拟合，所以实验二的预测精度不如实验一的预测精度。

这个案例研究的结果表明，这 7 个变量与产品中辛烷值是具有强相关性的，因此本研究中，选取原料性质中的芳烃、溴值和密度与两个待生吸附剂性质和两个再生吸附剂性质这 7 个变量作为问题三建模中的主要变量的一部分。

3.6 总结

在问题二的研究中，本研究首先通过原理分析，选择产品中的辛烷值作为选取问题三建模中的主要变量的关键因素。其次，通过对产品中辛烷值与其他变量的皮尔逊相关性分析和对产品中硫含量与其他变量的皮尔逊相关性分析得到了19 个主要变量，其中包含了 4 个原料性质和 15 个操作变量。最后，通过 BP 神经网络的预测模型验证了原料性质中的芳烃、溴值和密度与两个待生吸附剂性质和两个再生吸附剂性质这 7 个变量与产品中辛烷值是具有强相关性的，故再将它们选取为主要变量。综上所述，通过本研究设计的方法，在问题二中为问题三的模型建立选取了 26 个主要变量。

第四章 辛烷值预测模型建立与验证

在前言一节，我们分析了论文的整体框架。在数据预处理这一节，我们通过规定的方法和一些常规的方法对数据进行了预处理，剔除了一些采样数据异常的样本，得到了适用于做特征工程的数据。在数据特征工程阶段，我们通过利用皮尔逊相关性分析得出了跟所需输出结果相关性较大的操作变量。以下是辛烷值预测数学模型建立阶段，主要分为，1） 问题分析，通过对问题的深入分析，更好的选择预测模型。2） 算法简介，介绍问题分析这一节所拟采用的相关算法。3）模型建立与模型融合，利用特征工程所得到的数据进行预测模型建立与验证，然后通过选择预测精度更好的模型用于下一个问题的解决。4） 模型验证与结果分析，对模型进行横向纵向对比，验证我们所提出的模型的有效性。

4.1 问题分析

题目要求：采用数据预处理和降维得到的数据建立辛烷值损失预测模型，并进行模型验证。

通过对数据的预处理，我们利用最大最小限幅原则剔除了 14 个取值异常的数据样本，利用 3σ 原则剔除了 10 个原料性质异常的样本。所以我们现在能够使用的样本有 301 个。

通过上一步特征进行降维，我们得到的特征为：7 个原材料性质、2 个待生吸附剂性质、2 个再生吸附剂性质、15 个操作变量。考虑到每次精制汽油过程中原材料性质组成不由人的意志为转移，即使有些原材料性质与产品的相关性很低，为了保证模型结果的可操作性和适用性，我们并没有将相关性低的原料性质进行剔除。同样的道理，吸附性质也是在化工流程中无法忽略的特征，即使相关性较低，但是为了模型结果的可操作性与适应性，我们没有予以剔除。

从系统的角度来分析辛烷值损失预测问题，辛烷值损失既取决于原料性质、操作变量等一系列因素，又决定于产品的辛烷值。然而对于汽油精制系统来说，产品辛烷值是这个系统的一个输出，这个输出进一步影响着辛烷值损失，所以这是另一个因果系统。我们假定先对产品辛烷值进行预测，再通过预测结果对辛烷值损失进行考虑会更加合理，以下我们将通过模型建立与实验来验证我们的假定是否合理。

综合上述分析，该问题属于少样本学习问题，对于样本量较少的问题盲目使用深度学习方法容易产生过拟合，同时模型的稳定性也难以保证；另一方面，由于专业的限制，我们对于汽油精制流程的认识不够深入，可能会忽略一些相关性较弱但是十分必要的操作，从而导致我们选取的特征和剔除的样本之后的数据可能会存在或多或少的噪声点。所以我们首先排除了需要深度网络的相关算法。

近年来，机器学习算法的“市场份额”被深度学习占领了很大一个山头，但是毋庸置疑的是，对于样本量较少、数据噪声点多这一类常见的数据分析问题，传统机器学习算法的表现是深度学习算法无法比拟的。机器学习算法中我们选取了 XGBoost 、LightGBM、随机森林、SVM、BP 神经网络进行建模，通过对预测结果进行分析进行下一步处理。

4.2 算法简介

4.2.1 XGBoost 介绍

XGBoost 是陈天奇等人提出的一个改进机器学习算法[8]，高效地实现了 GBDT 算法并进行了算法和工程上的许多改进，可以说是 GBDT 算法工程上的一个实现。首先介绍 GBDT 算法。

GBDT 算法原理是指通过在残差减小的梯度方向建立 boosting tree（提升树），即 gradient boosting tree（梯度提升树）。每次建立新模型都是为了使之前模型的残差往梯度方向下降。

$$r_{ti} = -[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}] f(x) = f_{t-1}(x) \tag{4.1}$$

$$c_{tj} = \arg\min_{c} \sum_{x_i \in R_{tj}} L(y_i, f_{t-1}(xi) + c) \tag{4.2}$$

GBDT 缺点：GBDT 会累加所有树的结果，此过程无法通过分类完成，因为 GBDT 需要按照损失函数的梯度近似地拟合残差，这样拟合的是连续数据，因此只能是 CART 回归树，而不能是分类树。

XGBoost 属于集成学习 Boosting[9]，是在 GBDT 的基础上对 Boosting 算法进行的改进，并加入了模型复杂度的正则项。GBDT 是用模型在数据上的负梯度作为残差的近似值，从而拟合残差。XGBoost 也是拟合数据残差，并用泰勒展开式对模型损失残差的近似，同时在损失函数上添加了正则化项。

$$Obj^t = \sum_{i=1}^{n} L(y_i, \hat{y}_i^{t-1} + f_t(x_i)) + \Omega(f_t) + cons\tan t \tag{4.3}$$

其中 $\sum_{i=1}^{n} L(y_i, \hat{y}_i^{t-1} + f_t(x_i))$ 为损失函数，正则项包括如下两个部分：

$L1$ 正则化项：

$$L(\omega) = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 + \frac{\lambda}{2}\|\omega\|^1 \tag{4.4}$$

$L2$ 正则化项：

$$L(\omega) = \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 + \frac{\lambda}{2}\|\omega\|^2 \tag{4.5}$$

XGBoost 与 GBDT 算法的区别：

1、传统的 GBDT 在优化的时候只用到了一阶导数信息，而 XGBoost 则对代价函数进行了二阶泰勒展开，得到一阶和二阶导数，并且 XGBoost 在代价函数中加入了正则项，用于控制模型的复杂度。

2、另外 XGBoost 还支持线性分类器，通过在代价函数中加入正则项，降低了模型的方差，使学习出来的模型更加简单，避免过拟合。

以上是在一棵树的结构确定的情况下，求得每个叶子结点的分数。现在介绍如何确定树结构，即每次特征分裂怎么寻找最佳特征，怎么寻找最佳分裂点。

基于空间切分去构造一颗决策树是一个 NP 难题，我们不可能去遍历所有树结构，因此，XGBoost 使用了和 CART 回归树一样的想法，利用贪婪算法，遍历所有特征的所有特征划分点，不同的是使用上式目标函数值作为评价函数。具体做法就是分裂后的目标函数值比单子叶子节点的目标函数的增益，同时为了限制树生长过深，还加了个阈值，只有当增益大于该阈值才进行分裂。同时可以设置树的最大深度、当样本权重和小于设定阈值时停止生长去防止过拟合。

XGBoost 提出了两种防止过拟合的方法：Shrinkage and Column Subsampling。Shrinkage 方法就是在每次迭代中对树的每个叶子结点的分数乘上一个缩减权重 η，这可以使得每一棵树的影响力不会太大，留下更大的空间给后面生成的树去优化模型。Column Subsampling 类似于随机森林中的选取部分特征进行建树。其可分为两种，一种是按层随机采样，在对同一层内每个结点分裂之前，先随机选择一部分特征，然后只需要遍历这部分的特征，来确定最优的分割点。另一种是随机选择特征，则建树前随机选择一部分特征然后分裂就只遍历这些特征。一般情况下前者效果更好。

论文[10]中作者给的例子更容易理解 XGBoost 的原理，图 4-1 是一个回归树的目标函数计算的实例，Obj 代表了当指定一个树的结构的时候，在目标上面最多减少多少。即：Obj 代表一个数结构(模型)最大误差(损失)。

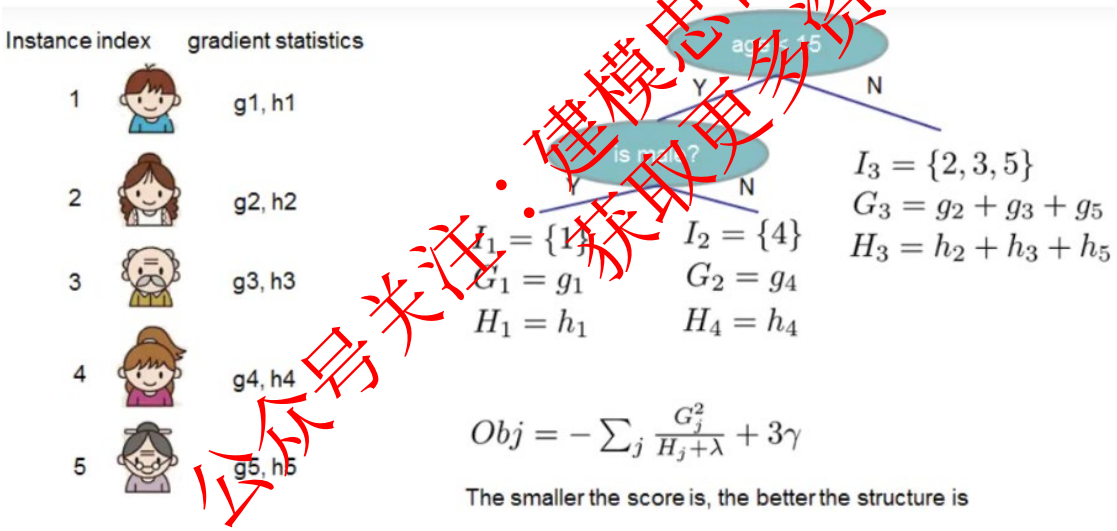


图 4-1 一个回归树的目标函数计算实例

树结构(模型)越拟合训练数据，分类误差(损失函数的值)就越小。在建树过程(建模)中每一次尝试去对已有的叶子进行分裂时，选用以下公式：

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma \tag{4.6}$$

其中，$\frac{G_L^2}{H_L+\lambda}$表示左子树分数，$\frac{G_R^2}{H_R+\lambda}$表示右子树分数，$\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}$表示不进行分割我们可以拿到的分数，$\gamma$表示加入新叶子节点引入的复杂度代价。

这样就可以在建树的过程中动态的选择是否要添加一个结点。特别注意：Obj 的值决定是否选择某个特征作为分裂结点。

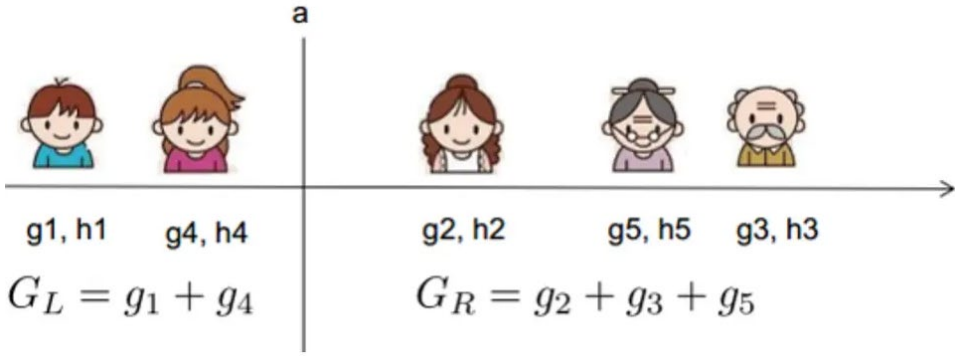


图 4-2 公式 Gain 的实例图

图 4-2 是公式 Gain 的实例，假设要枚举所有 x<a 这样的条件，对于某个特定的分割 a，要计算 a 左边和右边的导数和。对于所有的 a，我们只要做一遍从左到右的扫描就可以枚举出所有分割的梯度和 GL、GR。然后用上面的公式计算每个分割方案的分数就可以了。

不同的机器学习模型适用于不同类型的任务。深度神经网络通过对时空位置建模，能够很好地捕获图像、语音、文本等高维数据。而基于树模型的 XGBoost 则能很好地处理表格数据，同时还拥有一些深度神经网络所没有的特性（如：模型的可解释性、输入数据的不变性、更易于调参等）。这些特性使得 XGBoost 模型在深度学习广泛流行的今天没有被业界所抛弃而湮没在时代的潮流中。

4.2.2 LightGBM 介绍

LightGBM，它是微软出的新的 boosting 算法，基本原理与 XGBoost 一样，使用基于学习算法的决策树，只是在原有模型上做了一些优化（重点在模型的训练速度的优化）。在传统机器学习中，我们知道 XGBoost 算法非常热门，它是一种非常优秀的框架。但是在使用过程中，其训练耗时很长，内存占用比较大。LightGBM 相比于常规方法在不降低准确率的前提下，速度提升了 10 倍左右，占用内存下降了 3 倍左右。因为它是基于决策树算法的，采用最优的 leaf-wise 策略分裂叶子节点，然而其它的提升算法分裂树一般采用的是深度方向或者 level-wise 而不是 leaf-wise 的[11]。因此，在 LightGBM 算法中，当增长到相同的叶子节点，leaf-wise 算法比 level-wise 算法减少更多的损失。因此导致更高的精度，而其他的任何已存在的提升算法都不能够达。与此同时，它的速度也非常快，这就是该算法名字 light 的原因。

LightGBM 做出的改进主要是针对 XGBoost 的，所以针对 XGBoost 所做的相应改进为如下几部分：

1、LightGBM 基于 histogram 算法代替 pre-sorted 所构建的数据结构，利用 histogram 后，会有很多有用的 tricks。例如 histogram 做差，提高了 cache 命中率（主要是因为使用了 leaf-wise）。

2、在机器学习当中，我们面对大数据量时候都会使用采样的方式（根据样本权值）来提高训练速度。又或者在训练的时候赋予样本权值来关于于某一类样本（如 Adaboost）。LightGBM 利用了 GOSS 来做采样算法。

3、由于 histogram 算法对稀疏数据的处理时间复杂度没有 pre-sorted 好。因为 histogram 并不管特征值是否为 0。因此 LightGBM 采用了 EFB 来预处理稀疏数据。

LightGBM 使用的 Leaf-wise 策略：LightGBM 对于树的生长使用的是 Leaf-wise，而不是 Level-wise。主要是因为 LightGBM 认为 Level-wise 会产生一些低信息增益的节点，浪费运算资源[12]。通常来说，Level-wise 对于防止过拟合还是很有作用的，level-wise 与 Leaf-wise 相比就相形见绌了。Leaf-wise 能够追求更好的精度，让产生更好精度的节点做分裂。但这样带来过拟合的问题，所以一般使用的 max_depth 来控制它的最大高度。还有原因是因为 LightGBM 在做数据合并，Histogram Algorithm 和 GOSS 等各个操作[13]，其实都有天然正则化的作用，所以使用 Leaf-wise 来提高精度是一个很不错的选择。

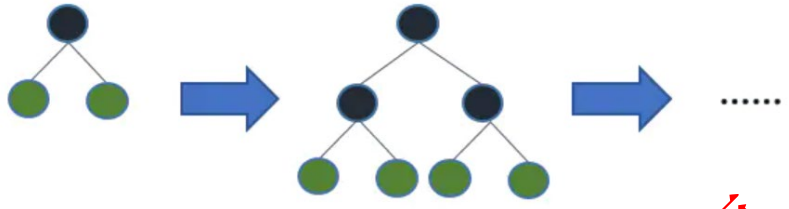


图 4-3 level-wise 策略下树的增长

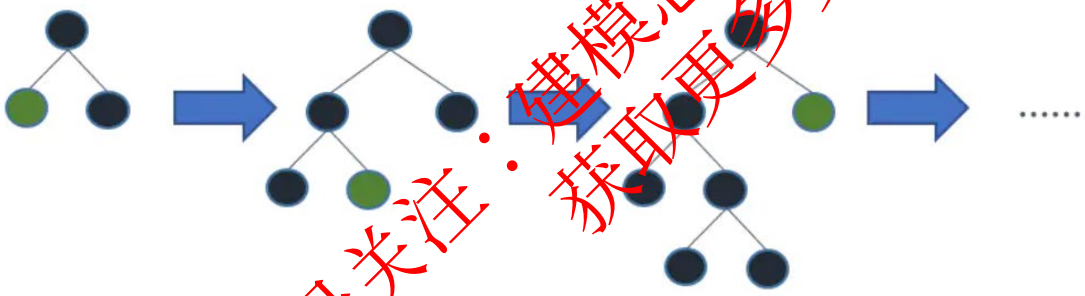


图 4-4 leaf-wise 策略下树的增长

从图 4-3 和图 4-4 可以看出 leaf-wise 策略下树的增长比 level-wise 策略的快很多，而且树的深度也变得更深了。相比于其他的机器学习方法，LightGBM 算法在保证准确率的前提下大大提升了计算速度，这也是我们选择这个算法的原因之一。

4.2.3 随机森林

随机森林算法是一种新兴起的机器学习算法。它拥有非常广泛的应用前景，从市场营销到医疗保健保险，既可以用来做市场营销模拟的建模，统计客户来源，保留和流失，也可用来预测疾病的风险和病患者的易感性。随机森林就是通过集成学习的思想将多棵树集成的一种算法，它的基本单元是决策树，而它的本质属于机器学习的一大分支——集成学习[14-16]（Ensemble Learning）方法。随机森林的名称中有两个关键词，一个是“随机”，一个就是“森林”。“森林”很好理解，一棵叫做树，那么成百上千棵就可以叫做森林了，这样的比喻还是很贴切的，其实这也是随机森林的主要思想--集成思想的体现。

其实从直观角度来解释，每棵决策树都是一个分类器（假设现在针对的是分类问题），那么对于一个输入样本，N 棵树会有 N 个分类结果。而随机森林集

成了所有的分类投票结果，将投票次数最多的类别指定为最终的输出，这就是一种最简单的 Bagging 思想。

随机森林中有许多的分类树。我们要将一个输入样本进行分类，我们需要将输入样本输入到每棵树中进行分类。打个形象的比喻：森林中召开会议，讨论某个动物到底是老鼠还是松鼠，每棵树都要独立地发表自己对这个问题的看法，也就是每棵树都要投票。该动物到底是老鼠还是松鼠，要依据投票情况来确定，获得票数最多的类别就是森林的分类结果。森林中的每棵树都是独立的，99.9%不相关的树做出的预测结果涵盖所有的情况，这些预测结果将会彼此抵消。少数优秀的树的预测结果将会超脱于芸芸“噪音”，做出一个好的预测。将若干个弱分类器的分类结果进行投票选择，从而组成一个强分类器，这就是随机森林 bagging 的思想（关于 bagging 的一个有必要提及的问题：bagging 的代价是不用单棵决策树来做预测，具体哪个变量起到重要作用变得未知，所以 bagging 改进了预测准确率但损失了解释性。）。图 4-5 可以形象地描述这个情况：

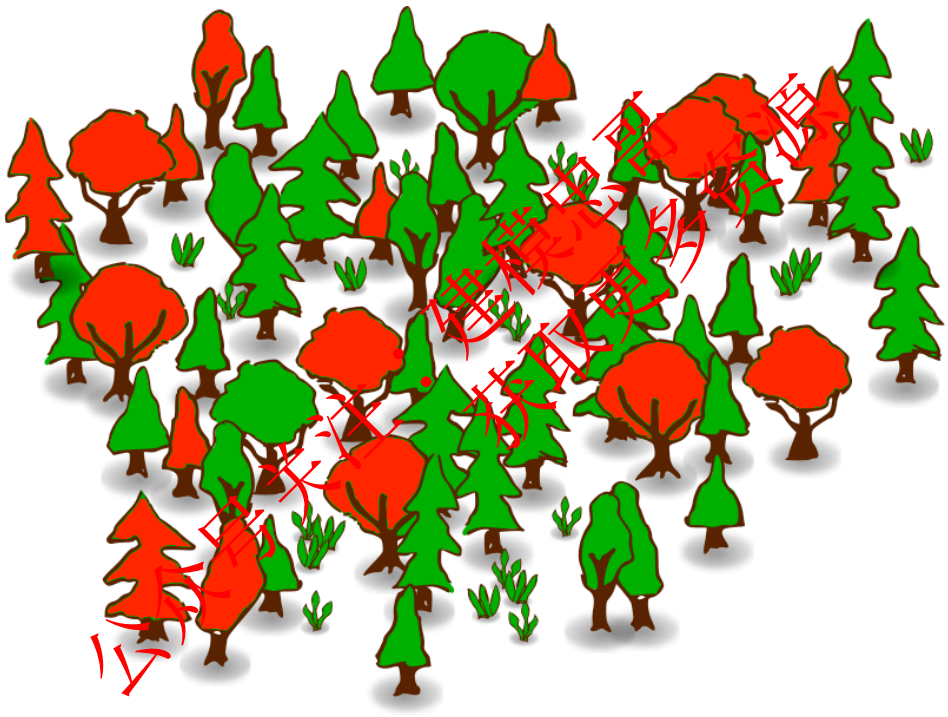


图 4-5 随机森林的形象描述

每棵树的按照如下规则生成：

1、如果训练集大小为 N，对于每棵树而言，随机且有放回地从训练集中的抽取 N 个训练样本（这种采样方式称为 bootstrap sample 方法），作为该树的训练集；

2、如果每个样本的特征维度为 M，指定一个常数 m<<M，随机地从 M 个特征中选取 m 个特征子集，每次树进行分裂时，从这 m 个特征中选择最优的；

3、每棵树都尽最大程度的生长，并且没有剪枝过程。

随机森林中的“随机”就是指的这里的两个随机性。两个随机性的引入对随机森林的分类性能至关重要。由于它们的引入，使得随机森林不容易陷入过拟合，并且具有很好得抗噪能力。

随机森林的优点可以概括如下：

1、训练可以高度并行化，对于大数据时代的大样本训练速度有优势；

2、能够处理很高维度（feature 很多）的数据，并且不用做特征选择；
3、可以用于特征选择，给出各个特征的重要性，缩减特征空间维度；
4、由于采用了随机采样，训练出的模型的方差小，泛化能力强；
5、实现简单，对部分缺失数据不敏感（由于是随机选择样本、随机选择特征）

但是它也有缺点，其缺点概括如下：

1、在噪声比较大的样本里随机森林模型容易陷入过拟合
2、对于有不同取值的属性的数据，取值划分较多的属性会对随机森林产生更大的影响

4.2.4 支持向量机

SVM(support vector machine)简单的说是一个分类器，并且是二类分类器。Vector：简单来说就是点，或是数据。Machine：也就是 classifier，也就是分类器。SVM 作为传统机器学习的一个非常重要的分类算法，它是一种通用的前馈网络类型，最早是由 Vladimir N.Vapnik 和 Alexey Ya.Chervonenkis 在 1963 年提出，目前的常见形式（soft margin）是 Corinna Cortes 和 Vapnik 在 1993 年提出，1995 年发表。深度学习（2012）出现之前，SVM 被认为是机器学习中近十几年最成功表现最好的算法。

支持向量机的浅显理解就是：给定训练样本，支持向量机建立一个超平面作为决策曲面，使得正例和反例的隔离边界最大化。

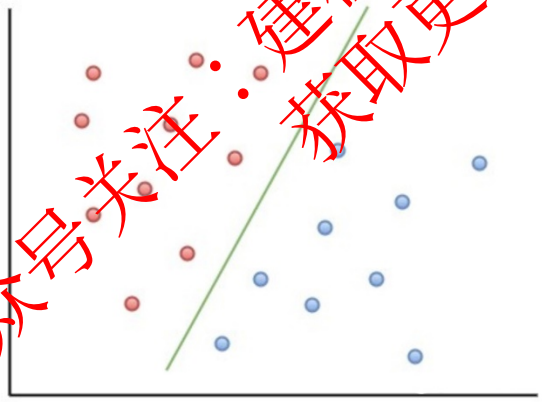
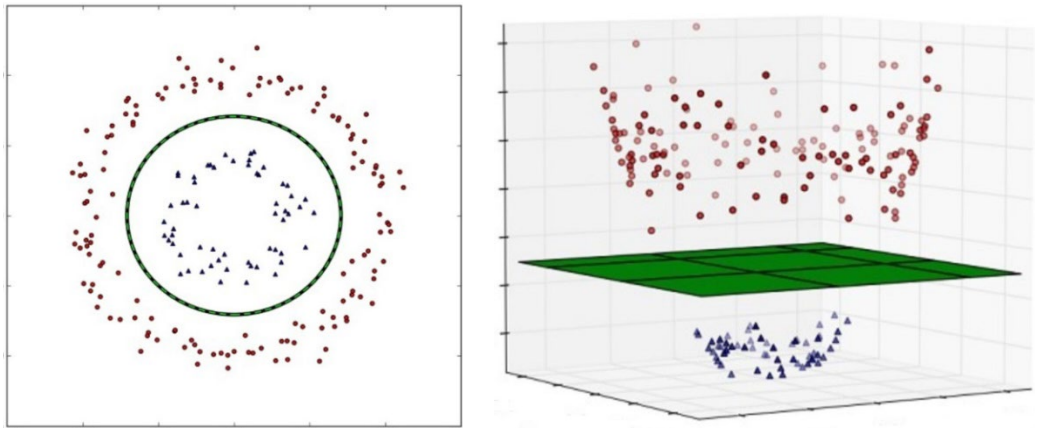


图 4-6 SVM 直观解释

从二维扩展到多维空间中时，将 D_0 和 D_1 完全正确地划分开的 $\omega x+b=0$ 就成了一个超平面。为了使这个超平面更具鲁棒性，我们会去找最佳超平面，以最大间隔把两类样本分开的超平面，也称之为最大间隔超平面。

最大间隔超平面：两类样本分别分割在该超平面的两侧、最大化两侧距离超平面最近的样本点到超平面的距离。

对于平面上的数据可以通过直观解释来理解，但是对于高维度的数据理解就相对复杂一些，高维数据一般是将他们映射到更高维度的空间去进行超平面的划分，图 4-7 是一个最大间隔超平面划分的例子，通过将图 4-7（a）中的数据映射到（b）中就能找到一个最大间隔超平面。



（a） (b)

图 4-7 SVM 实例

SVM 的优点如下：

1、有严格的数学理论支持，可解释性强，不依靠统计方法，从而简化了通常的分类和回归问题；

2、能找出对任务至关重要的关键样本（即：支持向量）；

3、采用核技巧之后，可以处理非线性分类/回归任务；

4、最终决策函数只由少数的支持向量所确定，计算的复杂性取决于支持向量的数目，而不是样本空间的维数，这在某种意义上避免了“维数灾难”。

同样的，SVM 的缺点如下：

1、训练时间长。当采用 SMO 算法时，由于每次都需要挑选一对参数，因此时间复杂度为$O(N^2)$，其中 N 为训练样本的数量；

2、当采用核技巧时，如果需要存储核矩阵，则空间复杂度为$O(N^2)$；

3、模型预测时，预测时间与支持向量的个数成正比。当支持向量的数量较大时，预测计算复杂度较高。

因此支持向量机目前只适合小批量样本的任务，无法适应百万甚至上亿样本的任务。由于我们的样本量只有几百条，特征数量也较多，所以 SVM 的这些特性使得它应用于我们的辛烷值预测任务是十分有效的。

4.2.5 BP 神经网络

BP 算法早在 20 世纪 80 年代就被提出了，它属于人工神经网络的一种，人工神经网络不用在预测前确定输入输出之间映射关系的具体的数学关系式，仅通过自身的训练，学习某种规则，在给定输入值时得到最接近期望输出值的结果。作为一种智能信息处理系统，人工神经网络实现其功能的核心是算法[17]。

基本 BP 算法包括信号的前向传播和误差的反向传播两个过程。即计算误差输出时按从输入到输出的方向进行，而调整权值和阈值则从输出到输入的方向进行。正向传播时，输入信号通过隐含层作用于输出节点，经过非线性变换，产生输出信号，若实际输出与期望输出不相符，则转入误差的反向传播过程。误差反传是将输出误差通过隐含层向输入层逐层反传，并将误差分摊给各层所有单元，以从各层获得的误差信号作为调整各单元权值的依据。通过调整输入节点与隐层节点的联接强度和隐层节点与输出节点的联接强度以及阈值，使误差沿梯度方向下降，经过反复学习训练，确定与最小误差相对应的网络参数(权值和阈值)，训

练即告停止。此时经过训练的神经网络即能对类似样本的输入信息，自行处理输出误差最小的经过非线形转换的信息[18]。

图 4-8 是 BP 神经网络的一个实例图，该网络有三个输入节点，四个隐藏节点，两个输出节点。

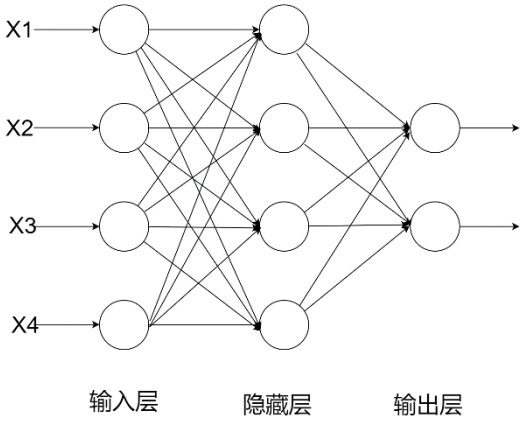


图 4-8 BP 神经网络拓扑结构

BP 神经网络算法执行的流程如下：

在手工设定了神经网络的层数，每层的神经元的个数，学习率 η（下面会提到）后，BP 算法会先随机初始化每条连接线权重和偏置，然后对于训练集中的每个输入 x 和输出 y，BP 算法都会先执行前向传输得到预测值，然后根据真实值与预测值之间的误差执行逆向反馈更新神经网络中每条连接线的权重和每层的偏好。在没有到达停止条件的情况下重复上述过程。

其中，停止条件可以是以下三条：

1、权重的更新低于某个阈值的时候

2、预测的错误率低于某个阈值

3、达到预设一定的迭代次数[18]

BP 神经网络的优点[19]**：**

1、非线性映射能力强

2、学习和自适应能力

3、泛化能力强

4、BP 神经网络具有一定的容错能力。

BP 神经网络的缺点也很明显：

1、容易陷入局部最优

2、BP 神经网络算法的收敛速度慢

3、应用实例与网络规模容易产生矛盾

4、BP 神经网络预测能力和训练能力容易产生矛盾

5、BP 神经网络对训练样本依赖性强

4.3 模型选择、建立与结果分析

这一部分我们分别利用 XGBoost、LightGBM、随机森林算法、支持向量机、BP 神经网络来建立汽油产品的辛烷值预测模型，再得到 $RON_{损失值}$：$RON_{损失值}$= $RON_{原料}$-$RON_{产品}$。然后分析模型预测效果，最后通过模型融合来集中模型之间的优点，在更短时间内获得更加精准预测效果。

Python 拥有各种集成机器学习工具包，我们分别使用了 python 的 sklear 库、xgboost、lightgbm 库中的 RandomForestRegressor、svm、XGBRegressor、LGBMRegressor 模块进行预测模型的建立，利用 pandas 库建立了 BP 神经网络预测模型。BP 神经网络预测模型参数：26 个输入节点、9 个隐藏单元、1 个输出节点。

通过样本筛选和特征提取，我们的样本量变成了 301 个，我们得到特征为：7 个原料性质、15 个操作变量、4 个吸附性质、1 个输出变量（RON 损失），以 9：1 的比例拆分数据集，在数据集中随机抽取 271 个用作训练，30 个样本用做预测，将进行特征工程之后的设计各个模块参数得到以下的结果：

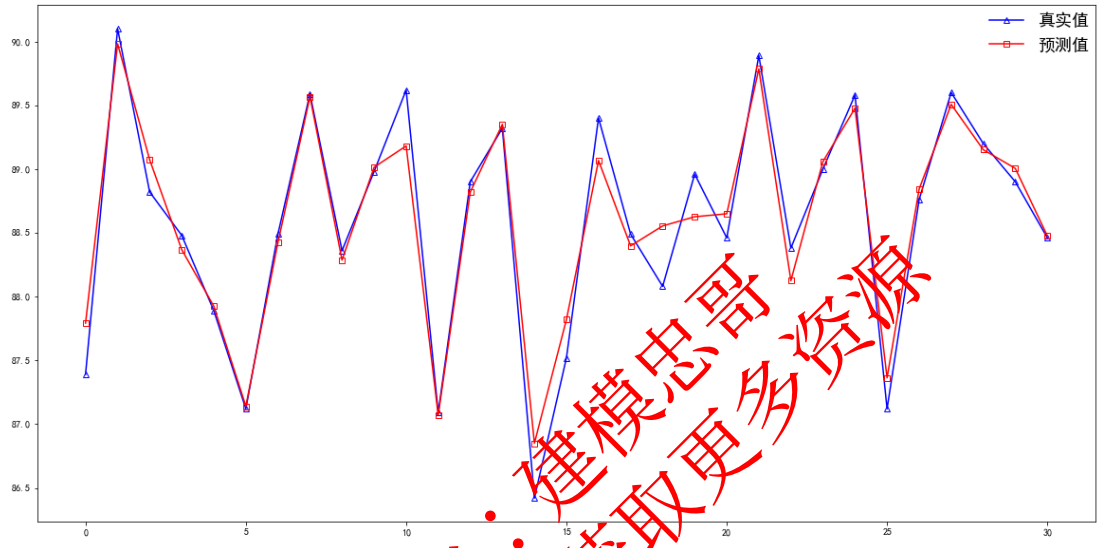


图 4-9 XGBoost 模型对产品辛烷值预测结果（测试样本 30 个）

如图 4-9，利用 XGBoost 模型对汽油辛烷值进行预测，总运行时间为 10.3s，通过计算，预测结果与真实值之间的均方根误差为 0.21335，平均绝对误差为 0.1604，平均绝对百分误差为 0.18147。通过分析预测结果的评价指标可以发现，XGBoost 能够很好的预测产品的辛烷值。但是在某些数据变化较大的点，模型的预测往往会更加突出，虽然总体精度较高，但是模型对数据的敏感性较高，一旦出现异常数据，可能模型就会失效。所以，XGBoost 对于样本量较少的任务往往存在模型稳定性较差的问题。

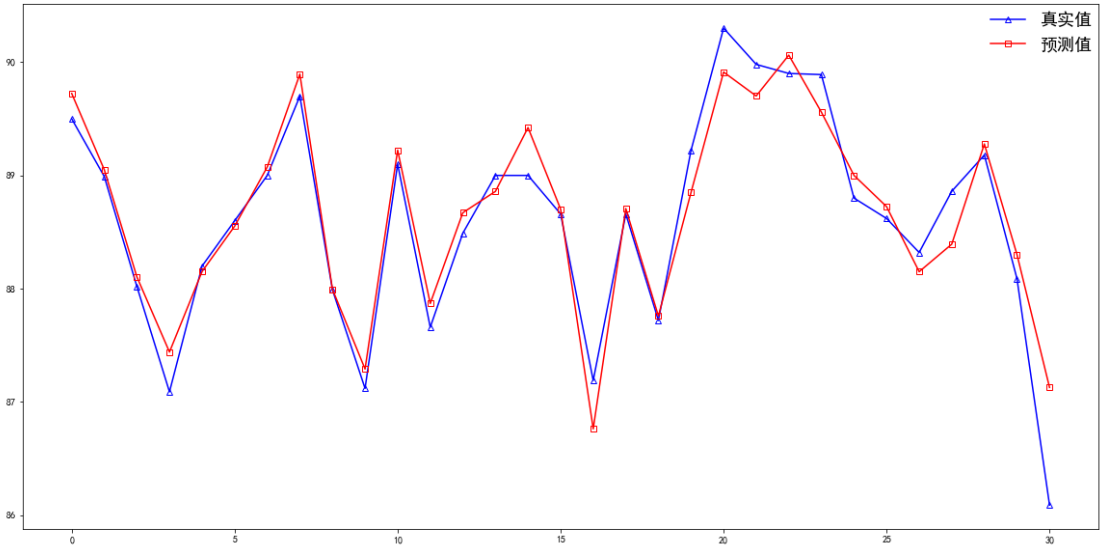


图 4-10 LightGBM 模型对产品辛烷值预测结果（测试样本 30 个）

如图 4-10，利用 LightGBM 模型对产品的 RON 进行预测。运行时间为 5.6s，速度非常快，这也是 LightGBM 模型的一个特点，通过计算，预测结果与真实值之间的均方根误差为 0.29363，平均绝对误差为 0.21663，平均绝对百分误差为 0.24521。通过分析预测结果的评价指标我们发现，预测的效果不如 XGBoost 模型精准，很多点的效果差距较大。虽然这个模型预测时间比较短，但是模型总体精准度较低，对于某一个样本的预测往往置信度较低。

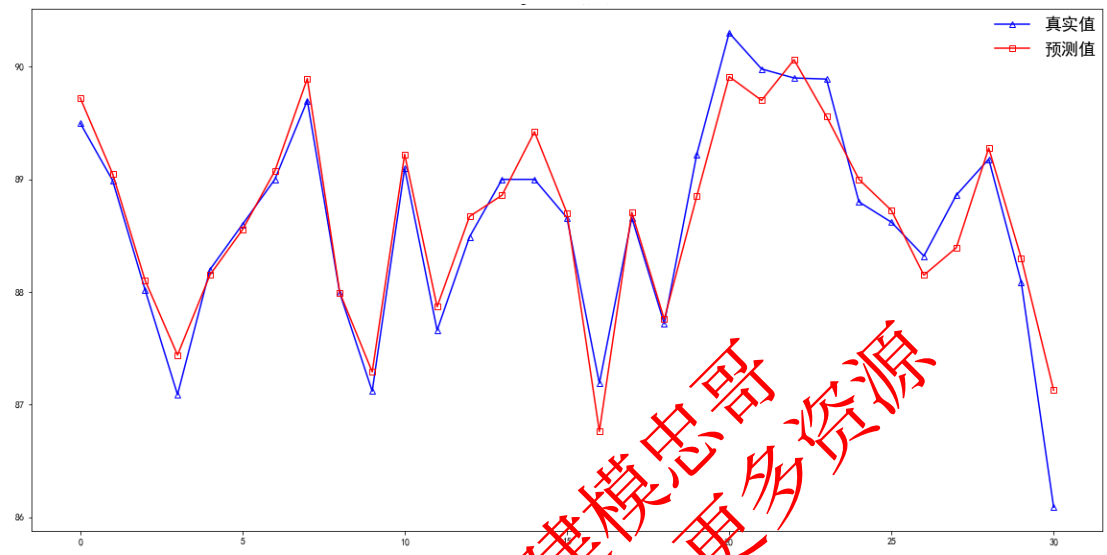


图 4-11 随机森林算法预测产品辛烷值结果图（测试样本 30 个）

如图 4-11，我们利用了随机森林算法对 RON 值进行了预测，运行时间大概为 19.9s，相较于前两种模型速度较慢，通过计算，预测结果与真实值之间的均方根误差为 0.21449，平均绝对误差为 0.15668，平均绝对百分误差为 0.17679。通过分析预测结果的评价指标我们发现随机森林算法预测结果较好，但是运行速度较慢，如果在工业环境运行，需要考虑硬件设备的的采样时间和软件外围环境的运行时间，对于随机森林算法的实际部署运行时间会更久。

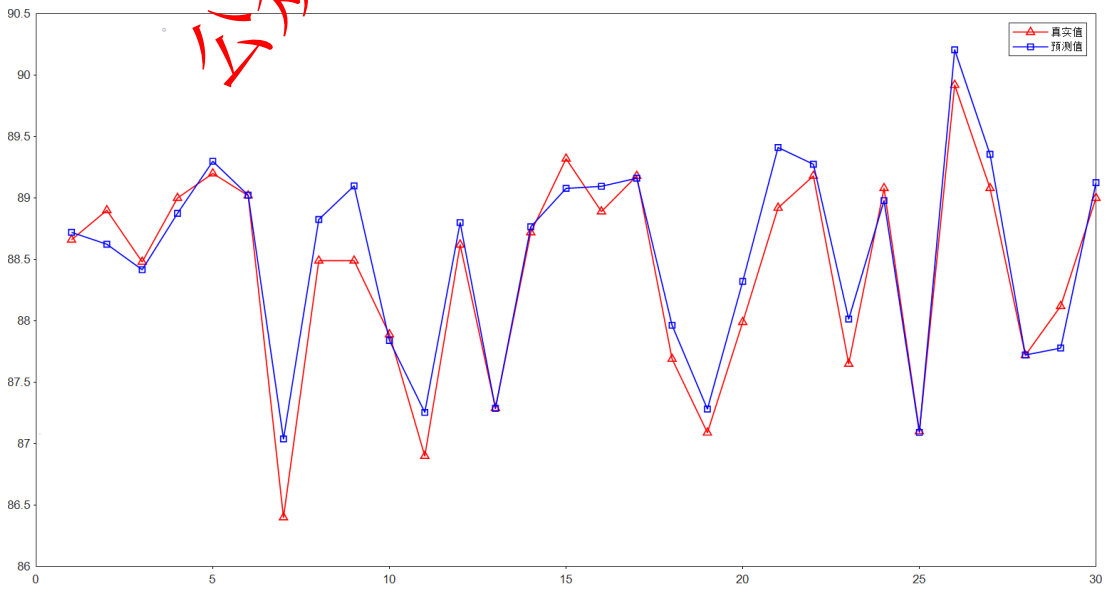


图 4-12 BP 神经网络预测产品辛烷值结果图

图 4-12，我们利用 BP 神经网络对产品辛烷值进行了预测分析，运行时间为

12s，通过计算产品辛烷真实值和产品辛烷预测的均方根误差 0.07239，平均绝对误差为 0.20665，平均绝对百分误差为 0.02343。通过分析预测结果的评价指标，我们发现 BP 神经网络在预测任务也能取得较好的结果，但是对于问题所描述的产品辛烷值预测任务中，数据样本量比较少，使用 BP 神经网络往往容易陷入过拟合、收敛速度较慢，这些缺点会模型适应能力会比较差。

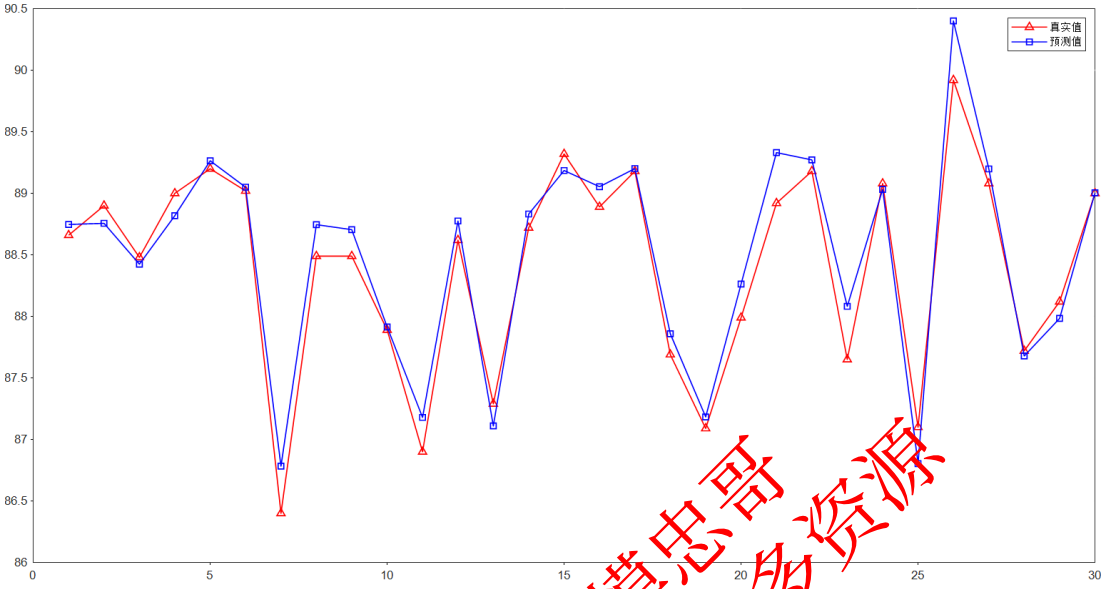


图 4-13 支持向量机预测产品辛烷值结果图

如图 4-13，我们利用了支持向量机对 RON 损失值进行了预测，运行时间大概为 9.4s，通过计算，得到了预测结果与真实值之间的均方根误差为 0.04512，平均绝对误差为 0.15611，平均绝对百分误差为 0.01924。通过分析预测结果的评价指标我们可以发现，支持向量机对于样本量较小的数据集预测精度非常好，预测误差分布也非常均匀，而且预测所花时间跟其他算法相差无几，很适合用于产品辛烷值的预测。

以上是我们所选择的各种模型及其实验结果，通过预测结果图能够看出基本上都能够对产品辛烷值变化规律进行拟合，但是具体的误差和误差分布情况我们需要进一步分析，下面通过我们选择的评价指标对模型预测误差进行逐一分析。

4.4 模型验证与对比

通过对模型的选择、模型建立和实验，我们得到了各种模型的评价指标，下面我们通过可视化的方式对所选择的各种模型进行分析。

表 4-1 各模型的评价指标

评价指标 / 模型	RMSE	MAE	MAPE
XGBoost	0.213353	0.160401	0.181472
LightGBM	0.293632	0.216629	0.245209
随机森林算法	0.214494	0.156684	0.176790
SVM	0.045102	0.156109	0.019241
BP 神经网络	0.0723863	0.206646	0.023428

图 4-14 是各种算法评价指标的可视化，通过可视化可以挖掘更多的模型预测信息，

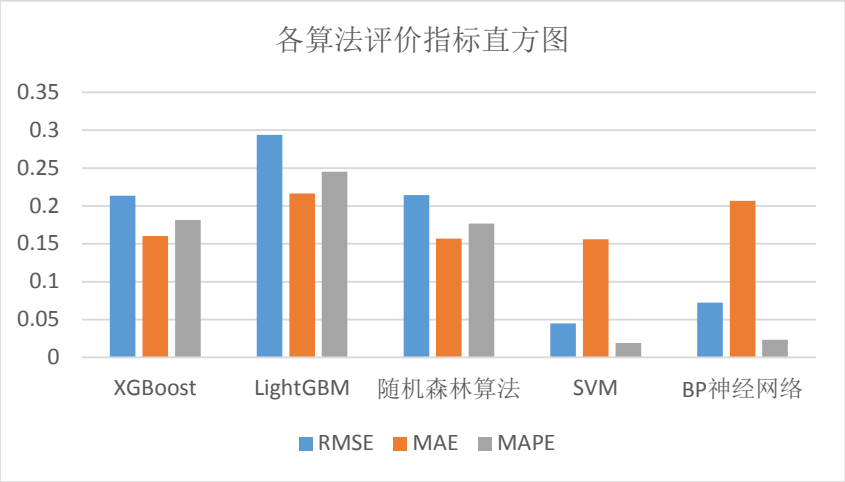


图 4-14 各算法评价指标直方图

通过对各种算法评价指标的分析，我们发现，关于均方根误差，SVM 算法和随机森林算法相较于其他算法较低，表示 SVM 算法和随机森林算法误差分布较为平均，相较于其他算法误差波动较为平稳；对于平均绝对误差，SVM 算法和 BP 神经网络相对于其他算法都较低，这就意味着 SVM 和 BP 神经网络能够很好的预测产品辛烷值的变化，预测结果于真实值之间的误差相对于其他算法要低；对于平均绝对百分比误差，SVM 和 BP 神经网络相比于其他算法都低得多，表示这两种模型所预测的结果中每个样本的平均误差均低于其他模型。进一步表明了 SVM 和 BP 神经网络在本题所述任务中的优越性。

对于 BP 神经网络，虽然效果较好，但是这种模型容易陷入局部最优，而且模型收敛速度很慢、导致训练时间比较长，而且模型结构只能通过经验确定，以上实验结果较好要归因于我们花费了大量时间进行模型参数调试。所以对于本题所给的任务，汽油精制过程的辛烷值损失预测，我们通过分析，辛烷值损失几乎都在 0-2 之间，绝对变化很小，但是辛烷值损失的相对变化较大，直接对辛烷值损失进行预测可能会导致模型敏感度较高，模型对各个相关变量变化的反应将会非常剧烈，预测辛烷值损失容易产生较大的误差。

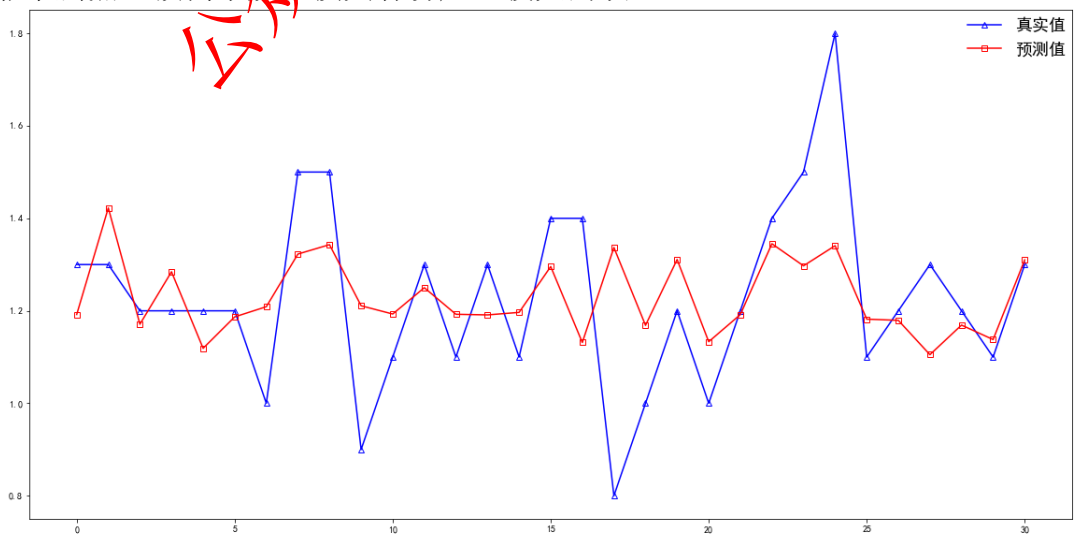


图 4-15 XGBoost 模型对辛烷值损失预测结果图

图 4-15 是我们初步利用 XGBoost 模型对特征选取可靠性所做的实验，图 4-16

是我们利用 XGBoost 模型对产品辛烷值进行的预测结果图。可以看出，辛烷损失值相对变化较大。反观产品辛烷值的变化，我们发现虽然它的绝对变化也较小，但是它的相对变化也比较小，对样本变量的变化将不会那么剧烈。

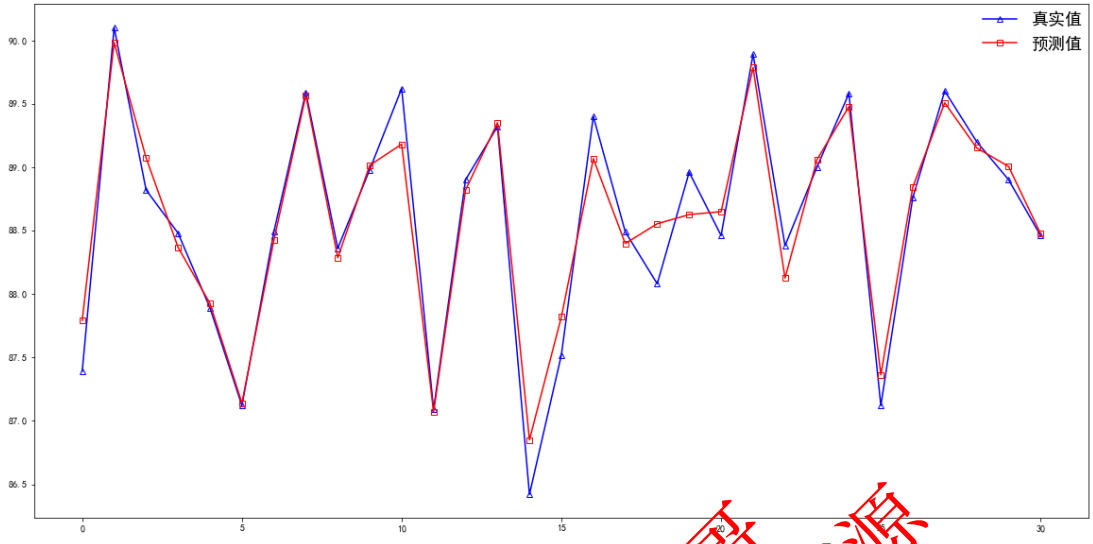


图 4-16 XGBoost 模型对产品辛烷值预测结果图

从系统的角度考虑，对于汽油精制系统来说，系统的输出是辛烷的剩余值，而非辛烷的变化值，辛烷变化值受到系统的输入和输出影响，这又是另一个因果系统，所以利用辛烷值因果系统对辛烷值损失值进行预测是不合理的，所以我们需要对产品辛烷值进行预测。

第五章　操作变量的优化方案

问题四要求我们建立操作变量的优化方案，实现辛烷值损失降幅最大化。在本节中，以辛烷值损失降幅最大为优化目标，主要变量中 15 个操作变量为决策变量，产品硫含量不大于 5μg/g 为其中一个约束条件，建立一个优化模型。通过粒子群算法、量子粒子群算法和差分量子粒子群算法对优化模型进行求解。

5.1 问题分析

问题四中，要求我们对主要变量中的所有操作变量进行优化，在产品硫含量小于等于 5μg/g 的条件下，寻找出操作变量的优化方案，使产品中的辛烷值损失降幅最大，并且在 301 个样本数据中筛选出辛烷值损失降幅大于等于 30%的样本。

这是一个典型的优化决策问题。因此，首先，建立决策变量、优化目标和约束条件的数学模型。然后根据建立好的优化模型，选择合适的优化算法对优化目标进行求解。

问题四中的优化目标是辛烷值损失降幅最大，这其中涉及了辛烷值的预测。因此，这一步需要调用问题三中已经训练好的 SVM 模型来预测与决策变量对应的辛烷值。同理，在约束条件中，对产品中的硫含量进行约束也涉及了产品中硫含量的预测，因此在本节中会在问题三种所建立的五种模型中，寻找最适合用于预测硫含量的模型。

群体智能算法的提出为优化问题提供了新的思路和解决方法，逐渐成为优化问题领域的一个研究热点。群体智能算法是一种启发式搜索算法，其寻优过程具有随机、并行以及分布式的特点，对于群体中的每个个体，其定义应根据实际求解问题的而定，并且不能保证每个个体在每个时刻都具有最佳的寻优特征，群体智能算法的特点体现在整个群体的总体优化特征。典型的群体智能算法有粒子群优化算法和蚁群优化算法，其中，粒子群优化算法适用范围广、寻优效果好和改良方法多。因此，本文在问题四中，选用粒子群优化算法来求解该优化模型。

为了提高优化的效果，在这里依次采用粒子群算法、量子粒子群算法和差分进化量子粒子群算法来对问题进行求解，从而筛选出辛烷值损失降幅大于等于 30%的样本。

5.2 优化模型的建立

在本节中，本研究将以辛烷值损失降幅最大为优化目标，主要变量中 15 个操作变量为决策变量，产品硫含量不大于 5μg/g 为其中一个约束条件，建立一个优化模型。然后通过优化算法求出辛烷值损失降幅最大时，最优的操作变量优化方案。

5.2.1 建立优化模型

(1) 决策变量的建立

将操作变量 S-ZORB.FT_1001.PV 、 S-ZORB.TE_1001.PV 、 … 、 S-ZORB.PC_1001A.PV 等 26 个设为决策变量 $x_1, x_2, \cdots, x_{15}$，记为 $X = [x_1, x_2, \cdots, x_{15}]$。

(2) 产品中辛烷值损失降幅模型的建立

通过问题三，利用训练好的 SVM 模型，求出决策变量 X 对应的产品中的辛烷值。不妨设产品中的辛烷值为 RON，可以由式(5.1)得到，

$$RON = F_1(X) \tag{5.1}$$

其中函数 $F_1(X)$ 表示训练好的 SVM 模型对决策变量 X 的拟合关系。

设原料中的辛烷值为 RON_0 和对应产品中实际的辛烷值 RON_1，则优化前辛烷值损失 RON_{LOSS0} 为

$$RON_{LOSS0} = RON_1 - RON_0 \tag{5.2}$$

优化后辛烷值损失 RON_{LOSS} 为

$$RON_{LOSS} = RON - RON_0 \tag{5.3}$$

由式(5.2)和(5.3)，可以得到产品中辛烷值损失降幅 f 为

$$f = \frac{RON_{LOSS} - RON_{LOSS0}}{RON_{LOSS0}} \tag{5.4}$$

(3) 产品中硫含量模型的建立

利用问题三中已经建立好的网络模型，对产品中的硫含量进行预测，通过研究发现，LGBM 对产品中含硫量的预测结果最好，如图 5-1 所示。利用训练好的 LGBM 模型，求出决策变量 X 对应的产品中的硫含量。

不妨设产品中的硫含量为 S，可以由式(5.5)得到，

$$S = F_2(X) \tag{5.5}$$

其中函数 $F_2(X)$ 表示训练好的 LGBM 模型对决策变量 X 的拟合关系。

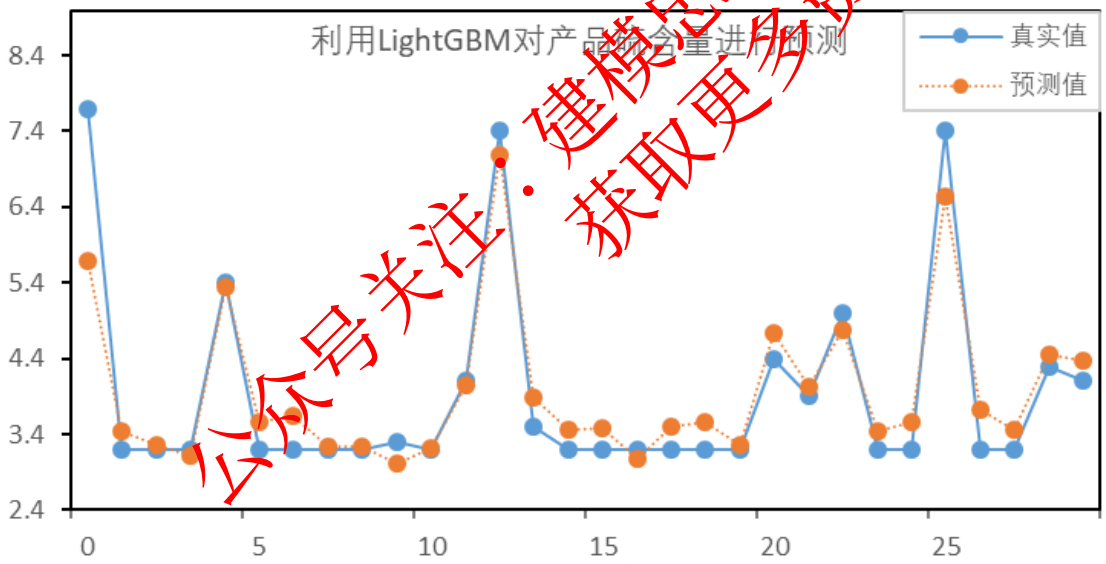


图 5-1 LGBM 对产品中含硫量的预测结果

(4) 建立目标函数

本研究以产品中辛烷值损失降幅 f 最大为优化目标，即式(5.6)

$$\max f \tag{5.6}$$

(5) 约束条件

① 决策变量的约束条件：

决策变量 $x_1, x_2, \cdots, x_{15}$ 应该满足约束条件(5.7)

$$x_{i\min} \le x_i \le x_{i\max} \tag{5.7}$$

其中，$x_{i\min}$ 和 $x_{i\max}$ 为操作变量 x_i 的上下限 $(i = 1, 2, \cdots, n)$。

② 产品中硫含量的约束条件：

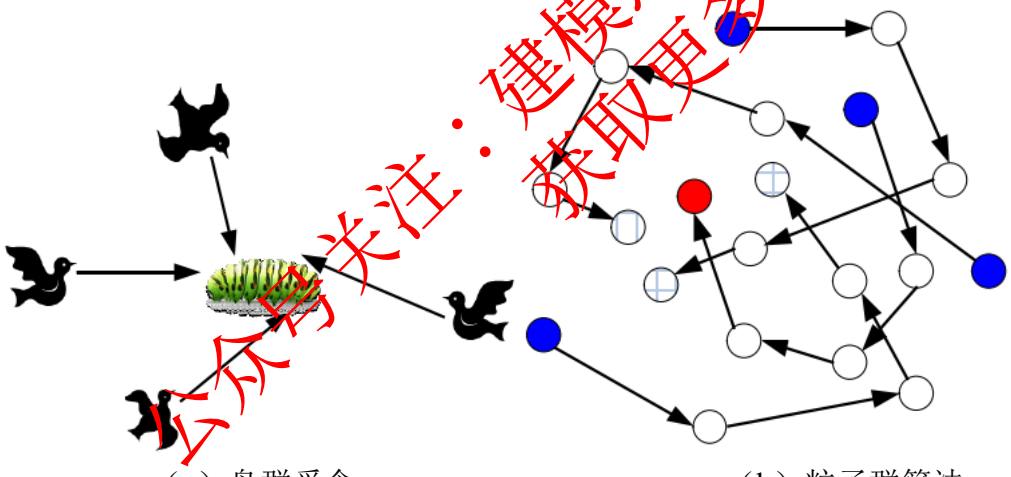
产品中硫含量 S 应该满足约束条件(5.8)

$$0 \le S \le 5 \tag{5.8}$$

5.3 粒子群算法的设计以及案例研究

粒子群优化（PSO）算法是 Kennedy 和 Eberhart 通过模拟群鸟寻食行为提出的，是一种新型的群体智能算法，可以用于解决优化问题。PSO 算法源于 Hepper 的对鸟群寻食行为的系统仿真，该系统的条件是每只鸟只知道自己和食物的距离但不清楚食物所在的方向，系统的基本思路是鸟群中的其他鸟总是朝着当前距离食物最近的鸟的方向飞行，直到找到食物。

PSO 算法便是从上述鸟群觅食的特性出发，加以数学化用以解决优化问题。PSO 系统中，每个优化问题的潜在解对应是鸟群中的一只鸟，PSO 将其描述为搜索空间（解空间）的一个粒子，每个粒子都有一个对应的适应度值，适应度值由具体优化问题的目标函数计算得到，对应的是鸟和食物之间的距离，适应度值的大小描述了粒子位置的好坏。每个粒子在问题的解空间中搜索最优解，粒子搜索过程中的运动速度由其本身的经验和群体经验动态决定。PSO 算法中，每个粒子整体的运动趋势是朝着当前适应度值最好的方向运动的，经过不断地迭代搜索，最终粒子群能够收敛到优化问题的最优解。每次迭代搜索中，粒子运动受粒子本身的惯性、以及两个最优位置影响，一个位置是粒子个体最优（personal best，pbest）位置，这是每个粒子目前位置找到的最优位置，另一个位置是全局最优（global best，gbest）位置，这也要求 PSO 中粒子应具备记忆功能，能够记住自身的个体最优位置，这是粒子群里面最佳粒子所处的位置。



（a）鸟群觅食　　（b）粒子群算法

图 5-2 鸟群觅食和粒子群算法

5.3.1 粒子群算法的设计

差分进化量子粒子群优化算法[20]、差分进化量子粒子群优化算法[21]

本研究首先采用基本的粒子群算法[22]，对上述建立的优化模型进行求解。

PSO 算法的数学描述为：设初始粒子群中有 M 个粒子，每个粒子处于 D 维的搜索空里面，第 n 次迭代过程中第 i 个粒子的位置为 $x_{i,n}=(x_{i,n}^{1},x_{i,n}^{2},.....,x_{i,n}^{D})$，该粒子对应的速度为 $v_{i,n}=(v_{i,n}^{1},v_{i,n}^{2},\cdots,v_{i,n}^{D})$。PSO 算法的迭代过程就是粒子不断更新其速度和位置的过程，而粒子 i 在第 n 次迭代中，速度和位置的更新公式分别为（5.9）和（5.10）：

$$v_{i,n+1}^{j}=v_{i,n}^{j}+c_1r_{i,n}^{j}\left(pbest_{i,n}^{j}-x_{i,n}^{j}\right)+c_2R_{i,n}^{j}\left(gbest_{n}^{j}-x_{i,n}^{j}\right) \quad (5.9)$$

$$x_{i,n+1}^{j}=x_{i,n}^{j}+v_{i,n+1}^{j} \quad (5.10)$$

（5.9）和（5.10）中，$i=1,2,\ldots M$ ，$j=1,2,\ldots D$；c_1 和 c_2 是加速度系数（acceleration coefficients），加速度系数调节粒子向 pbest 和 gbest 方向飞行的步长，反映了粒子受自身和和种群信息影响的程度，对平衡粒子的局部搜索能力和全局搜索能力起到重要的作用，通常取值为 2；$pbest_{i,n}=\left(pbest_{i,n}^{1},pbest_{i,n}^{2},\ldots,pbest_{i,n}^{D}\right)$ 是粒子在迭代过程中的最优位置，称为个体最优位置，$pbest_n=\left(pbest_n^{1},pbest_n^{2},\ldots,pbest_n^{D}\right)$ 是粒子群中，所有粒子的最优位置，称为全局最优位置。$r_{i,n}^{j}$ 和 $R_{i,n}^{j}$ 是介于 0 和 1 之间的随机数，即 $r_{i,n}^{j},R_{i,n}^{j}\sim U(0,1)$；通常情况下，需要对速度进行一定的限制，使得，$v_{i,n}^{j}\in[-v_{\max},v_{\max}]$，设计该阈值是为了更好地平衡搜索的性能，如果速度太大，可能会跳过最优解，太小又会搜索不充分。

从公式（5.9）可以看出，对于每个粒子（以第 i 个粒子为例），PSO 中粒子速度的更新主要包括三个部分：第一部分 $v_{i,n}$ 为粒子的运动惯性，也就是前一次迭代后粒子的速度；第二部分通常被称为个体认知（Cognition），代表粒子本身的运动经验，描述了粒子当前位置和自己的个体最优位置（ $pbest_{i,n}$ ）之间的距离： $pbest_{i,n}^{j}-x_{i,n}^{j}$；最后一部分是社会行为（Social），代表群体经验，是粒子当前位置和自己和全局最优位置之间的距离 $gbest_n^{j}-x_{i,n}$。

从上面的分析可以看出，粒子群中粒子的迭代过程主要是对速度和位置的更新，而速度的更新还受粒子的个体最优位置和群体最优位置的影响，因此，在速度和位置的更新的同时，还需要更新粒子的个体最优位置 pbest 和群体最优位置 gbest。假设一个最优化问题的描述如（5.11）所示：

$$\begin{aligned}&\min f(x)\\&s.tX\in S\subseteq R^{D}\end{aligned}\tag{5.11}$$

其中 $f(x)$ 是一个连续的目标函数，S 是最优解的可行空间，此时（5.11）为最小化问题。根据目标函数，个体最优 pbest 和全局最优 gbest 的更新公式为（5.12）和（5.13）：

$$pbest_{i,n}=\begin{cases}x_{i,n} & if \quad f(x_{i,n})<f(pbest_{i,n-1})\\ pbest_{i,n-1} & if \quad f(x_{i,n})\geq f(pbest_{i,n-1})\end{cases}\tag{5.12}$$

$$gbest_n=pbest_{g,n}\quad ,\quad g=\arg\min_{1\leq i\leq M}\left[f(pbest_{i,n})\right]\tag{5.13}$$

通过上述分析，基本粒子群算法流程如图 5-3 所示。

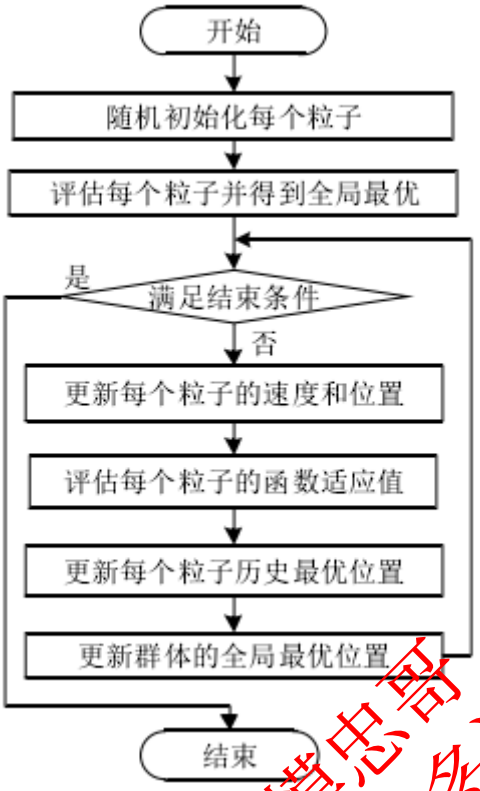


图 5-3　粒子群算法流程图

5.3.2 案例研究

本次研究中，将 301 个数据样本，依次通过优化算法进行求解。在 301 个样本中，只有 5 样本数据经过优化算法，在产品中硫含量 5μg/g 以下的约束下，这六个样本的辛烷值损失降幅达到了 30%以上，其余样本的辛烷值损失虽然经过优化得到了改善，但是无法使其辛烷值损失降幅达到了 30%以上。

表 5-1 通过优化算法晒选出来的 6 个样本的具体情况

特征 \ 样本编号	62	76	135	151	153
RON_{LOSS0}	1.8	1.8	1.8	1.8	1.8
RON_{LOSS}	1.2167	1. 2462	1.1771	1.1962	1.2506
辛烷值损失降幅	32.41%	30.77%	34.61%	33.54%	30.52%
产品硫含量	3.8%	4.3%	4.7%	4.1%	3.3%

如表 5-1 所示，由结果可以知道，此次的优化通过的样本的原辛烷值损失都是 1.8，而其他样本的原辛烷值损失小于 1.8 的都没有通过优化实现辛烷值损失降幅大于 30%。这有可能式由于粒子群优化算法的局限性，造成了算法陷入了局部最优，从而无法得到很好的优化结果。为此，针对这一情况，我们将对基本的粒子群优化算法进行改进优化。

5.4 量子粒子群算法的设计以及案例研究

为了克服基本粒子群算法容易陷入局部最优的缺陷，我们对其进行了改进，根据前人的研究针对本文的问题，设计量子粒子群算法[23]。

5.4.1 量子粒子群算法的设计

粒子在量子系统中的速度和位置不能同时确定，粒子的状态是由波函数ψ描述的，量子空间中，$|\psi|^2$表示粒子状态的概率密度函数，粒子是在以局部吸引子为中心的δ势阱中迭代更新，最终达到收敛。假设粒子群体处于 D 维的量子空间，在第n次迭代中，第i个粒子在以局部吸引点p_i为中心的δ势阱中运动。在第$n+1$次迭代中，设$Y_{i,n+1}^j=|x_{i,n+1}^j-p_{i,n}^j|$，则对应的波函数为：

$$\psi\left(Y_{i,n+1}^j\right)=\frac{1}{\sqrt{L_{i,n}^j}}\exp\left(-\frac{Y_{i,n+1}^j}{L_{i,n}^j}\right)\quad,\quad\left(1\le i\le M,1\le j\le D\right)\tag{5.14}$$

进一步可以得到粒子状态的概率密度函数 Q ：

$$\begin{aligned}Q\left(Y_{i,n+1}^j\right)&=|\psi\left(Y_{i,n+1}^j\right)|^2\\&=\frac{1}{\sqrt{L_{i,n}^j}}\exp\left(-\frac{2Y_{i,n+1}^j}{L_{i,n}^j}\right)\end{aligned}\tag{5.15}$$

相应的概率分布函数F：

$$F\left(Y_{i,n+1}^j\right)=1-\exp\left(-\frac{2Y_{i,n+1}^j}{L_{i,n}^j}\right)\tag{5.16}$$

其中，$L_{i,n}^j$是双指数分布的标准偏差，描述了波函数的特征长度，也就是粒子出现在相对位置的概率。最后应用蒙特卡洛方法，得到第$n+1$迭代后粒子的状态：

$$x_{i,n+1}^j=p_{i,n}^j\pm\frac{L_{i,n}^j}{2}\ln\left(1/u_{i,n}^j\right),\quad u_{i,n}^j\sim\mathrm{U}(0,1)\tag{5.17}$$

其中，$u_{i,n}^j$是在（0，1）区间上均匀分布的随机数，而$L_{i,n}^j$的计算如下：

$$L_{i,n}^j=2\alpha_n|x_{i,n}^j-C_n^j|\tag{5.18}$$

（5.18）中$C_n=\left(C_n^1,C_n^2,\ldots C_n^D\right)$是平均最优位置，即所有粒子的个体最优的均值，计算方式如下：

$$C_n^j=\frac{1}{M}\sum_{i=1}^{M}pbest_{i,n}^j\tag{5.19}$$

结合（5.17）和（5.18），可以得到 QPSO 中，粒子状态更新方程为：

$$x_{i,n+1}^j=p_{i,n}^j\pm\alpha_n|x_{i,n}^j-C_n^j|\ln\left(1/u_{i,n}^j\right)\tag{5.20}$$

式中，参数α为压缩-扩张因子，该参数用于平衡迭代过程中粒子的局部和全局搜索性。

综上所述，QPSO [24]和 PSO 的在算法思想上是一致的，区别在于两中算法中粒子的运动模型，QPSO 中粒子状态的更新过程如下：

$$C_n=\left(C_n^1,C_n^2,\ldots C_n^D\right)=\left(\frac{1}{M}\sum_{i=1}^{M}pbest_{i,n}^1,\frac{1}{M}\sum_{i=1}^{M}pbest_{i,n}^2,\ldots\frac{1}{M}\sum_{i=1}^{M}pbest_{i,n}^D\right)\tag{5.21}$$

$$p_{i,n}^j=\varphi_{i,n}^j pbest_{i,\mathrm{n}}^j+\left(1-\varphi_{i,n}^j\right)gbest_{i,\mathrm{n}}^j\tag{5.22}$$

$$x_{i,n+1}^j=p_{i,n}^j\pm\alpha_n|x_{i,n}^j-C_n^j|\ln\left(1/u_{i,n}^j\right)\tag{5.23}$$

QPSO 中，在粒子状态更新结束后，同样需要对粒子个体最优状态 ipbest 和群体最优状态 gbest 进行更新，ipbest 和 gbest 的更新方式和 PSO 相同，即

$$pbest_{i,n} = \begin{cases} x_{i,n} & if \quad f\left(x_{i,n}\right) < f\left(pbest_{i,n-1}\right) \\ pbest_{i,n-1} & if \quad f\left(x_{i,n}\right) \geq f\left(pbest_{i,n-1}\right) \end{cases} \tag{5.24}$$

$$gbest_n = pbest_{g,n} \quad , \quad g = \arg\min_{1 \leq i \leq M} \left[f\left(pbest_{i,n}\right) \right] \tag{5.25}$$

5.4.2 案例研究

与之前的案例研究方法相似，将 301 个数据样本，依次通过量子粒子群算法进行求解。

由于量子粒子群算法相比于基本的粒子群算法具有更好的搜索能力和更大搜索空间，因此，这次在 301 个样本中有 21 个样本数据经过优化算法，在产品中硫含量 5μg/g 以下的约束下，这 21 个样本的辛烷值损失降幅达到了 30%以上。这 21 个样本的编号为 15、16、31、59、61、62、70、71、76、79、84、101、107、108、135、138、149、151、173 和 303。

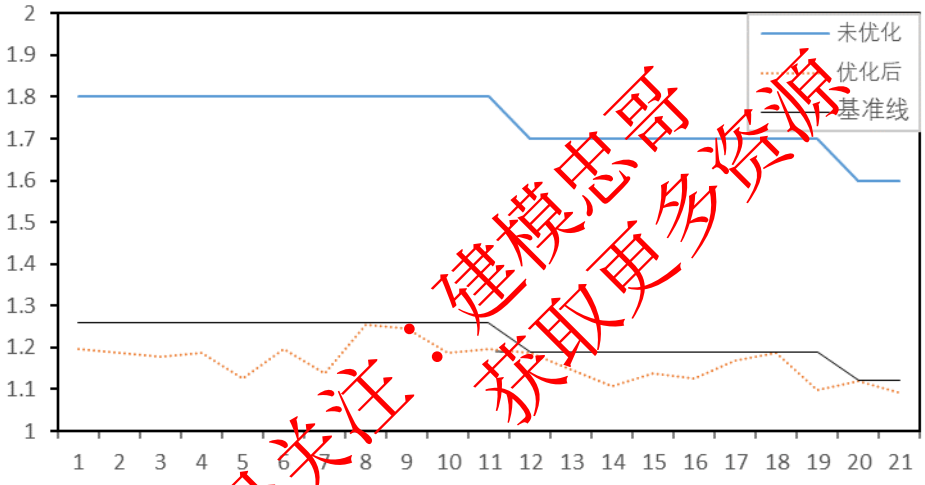


图 5-4 21 个样本辛烷值损失变化曲线

这 21 个样本辛烷值损失变化曲线如图 5-4 所示。由图可知，只要辛烷值损失在基准线下方，就样本的辛烷值损失降幅就达到了 30%及其以上。通过优化模型得到的样本占总样本的 6.98%，如图 5-5 所示。

由此次实验的结果可以证明，量子粒子群算法的寻优效果由于基本的粒子群优化算法。但是从总体来说，被优化的样本主要是原来辛烷值损失大于等于 1.7 的样本，其中两个辛烷值损失为 1.6 的样本也通过了优化。这也说明了量子粒子群算法也无法令人得到满意的答案，需要我们对算法进行更加深入的改进。

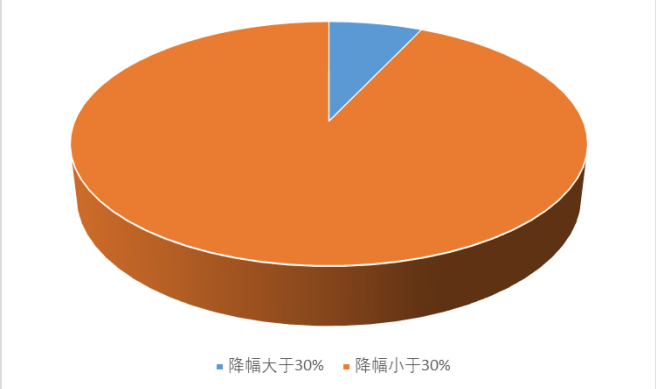


图 5-5 辛烷值降幅大于 30%的比例

5.5 差分进化量子粒子群算法的设计以及案例研究

QPSO 算法克服了普通 PSO 算法在收敛性上的不足：

首先，QPSO 系统所在的量子系统是非线性的复杂系统，并且满足状态重叠原理，这样粒子在 QPSO 中比在 PSO 中具有更多的状态；其次，QPSO 算法在运行时，粒子的搜索范围是整个可行解空间，粒子是以一定的概率出现在搜索范围里的任意位置，而无法用具体的轨迹描述；最后，由于 PSO 的搜索范围有限，粒子也被限制在搜索空间的某一个局部位置，而 QPSO 中由于搜索空间是整个可行解空间，因此 QPSO 的结果具有更好的适应值。

然而 QPSO 在运行过程中，尽管粒子的搜索空间很大，但粒子还是会逐渐的聚集并收敛到最优解，在后期的搜索中，粒子逐渐聚集，粒子种群的多样性也会迅速降低，若此时收敛到的结果是一个局部最优解，种群多样性的丧失使得粒子群体失去了拓展搜索空间的能力，很可能使得算法陷入局部最优。而这也是使用了 QPSO 得到的优化结果仍然不如人意的原因之一。

不过，差分进化（DE）算法中的变异操作有助于增加全局搜索能力，保证了一定的种群多样性。为了有效利用 DE 算法和 QPSO 的优点，同时将 DE 思想引入到 QPSO 中以弥补 QPSO 在搜索后期多样性的不足，本研究对 QPSO 进行再次改进，根据相关文献资料设计基于差分进化的量子粒子群算法。

5.5.1 差分进化量子粒子群算法的设计

为了将 DE 思想引入到 QPSO 中，需要重新设计 QPSO 的状态更新模型，DE-QPSO 以 QPSO 算法为基本框架，在粒子更新过程中添加基于 DE 思想的变异、交叉选择操作。假设问题模型和 5.4.1 节中的内容相同，DE-QPSO 中，粒子进行状态更新时，不直接通过吸引子加势阱形式实现，而是在此基础上加上一个扰动以生成一个变异粒子 $v_{i,n+1}$，描述如下：

$$v_{i,n+1}^{j} = p_{i,n}^{j} \pm \alpha_n \left| x_{i,n}^{j} - C_n^{j} \right| \ln\left(1/u_{i,n}^{j}\right) + F\left(x_{r1,n}^{j} - x_{r2,n}^{j}\right) \tag{5.26}$$

变异粒子的实现时通过两部分的组合实现的，第一部分是基于 QPSO 中的局部吸引子加上 δ 势阱，这部分充分利用了 QPSO 种粒子在整个搜索空间进行搜索的优势，具有很好的全局搜索性能，保证了全局收敛性。第二部分是差分向量，为粒子状态增加了一个扰动，提高了种群的多样性。其中 F 是为缩放因子，为了简化算法的参数选择，这里采用固定值 0.5。

生成变异粒子 $v_{i,n+1}$ 之后，接着对变异粒子进行交叉操作（5.25），对（5.26）产生的变异个体 $v_{i,n+1}$ 进行交叉操作，生成试验粒子 $u_{i,n+1}$：

$$u_{i,n+1}^{j} = \begin{cases} v_{i,n+1}^{j} & ,if\left(rand(0,1) \le CR \quad or \quad j = jrand\right) \\ x_{i,n}^{j} & ,otherwise \end{cases} \tag{5.27}$$

其中，CR 是交叉概率，介于（0，1），为了简化算法，这里本文使用固定值0.8，$jrand$ 是一个随机数。满足 $jrand \in [1,2,3\ldots D]$。

接着进行选择操作，DE-QPSO [20]中选择操作和DE算法中的选择操作基本相同：

$$x_{n+1} = \begin{cases} u_{i,n+1} & ,if\left(f\left(u_{i,n+1}\right) \le f\left(x_{i,n}\right)\right) \\ x_{i,n} & ,otherwise \end{cases} \tag{5.28}$$

至此，粒子状态更新结束，DE-QPSO算法框架中，最后还需要更新粒子个体最优 $pbest_{i,n}$ 和群体最优 $gbest_{i,n}$：

$$pbest_{i,n}=\begin{cases}x_{i,n} & if \quad f\left(x_{i,n}\right)<f\left(pbest_{i,n-1}\right)\\ pbest_{i,n-1} & if \quad f\left(x_{i,n}\right)\geq f\left(pbest_{i,n-1}\right)\end{cases} \tag{5.29}$$

$$gbest_n=pbest_{g,n} \quad , \quad g=\arg\min_{1\leq i\leq M}\left[f\left(pbest_{\mathrm{i},n}\right)\right] \tag{5.30}$$

从上面的分析可以看出，DE-QPSO [21]本质上是在 QPSO 算法的框架下引入了 DE 的思想进行粒子的状态更新，这样做的目的是借助 DE 算法的变异、交叉和选择的操作改善 QPSO 在迭代过程中种群多样性的减少，同时又不失 QPSO 全局收敛的特性，提高了算法的性能。最后，重新归纳下 DE-QPSO 的算法基本流程：

Begin：

初始化每个粒子的当前状态 $\left\{x_{i,0}\right\}_{i=1}^{M}$，计算每个状态对应的适应度值，更新粒子个体最优 $\left\{pbest_{i,0}\right\}_{i=1}^{M}$ 和全局最优 $gbest_{i,0}$，设置当前迭代次数 $n=0$；

While（不满足算法结束条件）

Do

$n=n+1$；

根据（5.21）计算平均最优位置 C_n；

选择合适的 α；

for i =1: M

for j =1: D

$\varphi_{i,n}^{j}=rand1(.)$；

计算吸引子 $p_{i,n}^{j}=\varphi_{i,n}^{j}pbest_{i,\mathrm{n}}^{j}+\left(1-\varphi_{i,n}^{j}\right)gbest_{\mathrm{n}}^{j}$；

$u_{i,n}^{j}=rand2(.)$；

if rand3(.)<0.5

（5.26）进行变异操作：

$$v_{i,n+1}^{j}=p_{i,n}^{j}+\alpha_n\left|x_{i,n}^{j}-C_n^{j}\right|\ln\left(1/u_{i,n}^{j}\right)+F\left(x_{r1,n}^{j}-x_{r2,n}^{j}\right)$$

els

（5.26）进行变异操作：

$$v_{i,n+1}^{j}=p_{i,n}^{j}-\alpha_n\left|x_{i,n}^{j}-C_n^{j}\right|\ln\left(1/u_{i,n}^{j}\right)+F\left(x_{r1,n}^{j}-x_{r2,n}^{j}\right),$$

end

根据(5.27)进行交叉操作：

$$u_{i,n+1}^{j}=\begin{cases}v_{i,n+1}^{j} & ,if\left(rand\left(0,1\right)\leq CR \quad or \quad j=jrand\right)\\ x_{i,n}^{j} & ,otherwise\end{cases},$$

根据(5.28)进行选择操作：$x_{n+1}=\begin{cases}u_{i,n+1} & ,if\left(f\left(u_{i,n+1}\right)\leq f\left(x_{i,n}\right)\right)\\ x_{i,n} & ,otherwise\end{cases}$,

end f

根据目标函数，计算粒子的适应度值 $f\left(x_{i,n}\right)$，并依据公式（5.29）和（5.30）

更新个体最优 $pbest_{i,n}$ 和全局最优 $gbest_n$；

end f
end do
End

5.5.2 案例研究

与前两次的案例研究方法相似，将 301 个数据样本，依次通过差分进化量子粒子群算法进行求解。

由于差分进化量子粒子群算法克服了量子粒子群陷入局部最优的缺陷，因此，这次在 301 个样本中有 41 个样本数据经过优化算法，在产品中硫含量 5μg/g 以下的约束下，这 41 个样本的辛烷值损失降幅达到了 30%以上。

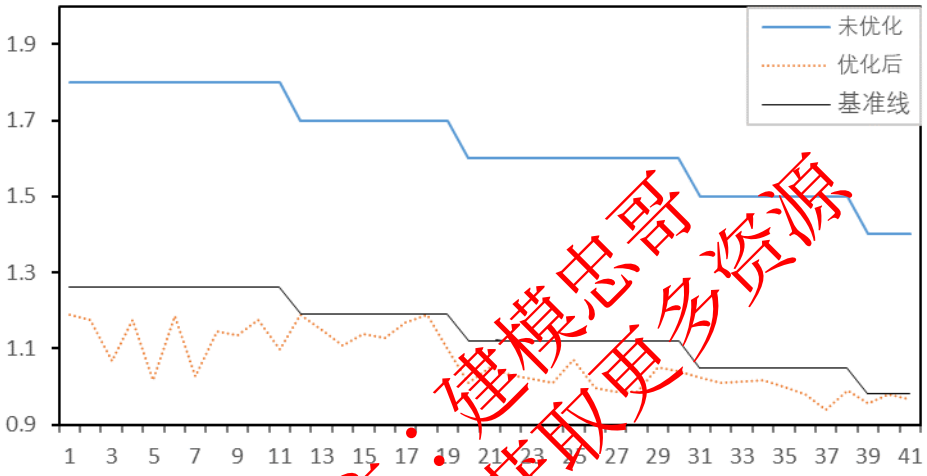


图 5-5 41 个样本辛烷值损失变化曲线

这 41 个样本辛烷值损失变化曲线如图 5-5 所示。由图可知，只要辛烷值损失在基准线下方，就样本的辛烷值损失降幅就达到了 30%及其以上。通过优化模型得到的样本占总样本的 13.62%，如图 5-6 所示。

通过引进差分进化，增加了量子粒子群算法的全局搜索能力，使得这次的优化结果对比前两次的优化结果有了明显的提升。从总体来看，被优化的样本包含了全体样本中原来辛烷值损失为 1.8、1.7、1.6 和 1.5 的所有样本，并且有三个原来辛烷值损失为 1.4 的样本也通过了优化。

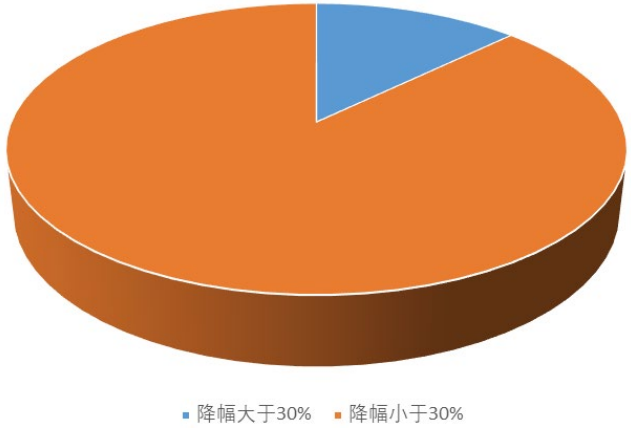


图 5-6 辛烷值降幅大于 30%的比例

但是，从整个算法改进的优化过程来看，优化的结果还是无法让人满意，尤

其时原来辛烷值损失小于等于 1.4 的样本，基本上很难被优化。所以，这可能不仅仅是优化算法造成的原因，还有可能是本节中模型建立的不够科学或者 SVM 和 LGBM 的拟合精度还不够高所导致我们的优化结果仍然较差。希望在后续的研究中可以这些问题进行深入的优化。

5.6 总结

在本节中，以辛烷值损失降幅最大为优化目标，主要变量中 15 个操作变量为决策变量，产品硫含量不大于 5μg/g 为其中一个约束条件，建立一个优化模型。通过粒子群算法、量子粒子群算法和差分量子粒子群优化算法对优化模型进行求解，从 301 个样本数据中筛选出辛烷值损失降幅大于等于 30%的样本。

在依次对优化模型使用粒子群算法、量子粒子群算法和差分量子粒子群算法来求解的过程中可以发现，随着算法的改良，从 301 个样本数据中筛选出辛烷值损失降幅大于等于 30%的样本数量也在增长，其中差分量子粒子群算法的优化效果最为显著。但是，总体上的优化效果和筛选样本效果都不够好，不能令人满意。这其中的原因不仅仅是算法自身的缺陷，也有可能是本节中模型建立的不够科学或者 SVM 和 LGBM 的拟合精度还不够高，希望在后续的研究中可以针对该问题进行深入的研究。

第六章 模型可视化展示

6.1 问题分析

工业装置为了平稳生产，优化后的主要操作变量往往只能逐步调整到位。上一章我们利用差分进化量子粒子群优化算法对操作变量进行寻优并且得到了主要操作变量的最优操作条件。本章通过对 133 号样本的主要操作变量进行逐步调整，绘制出了达到最优操作条件时的辛烷值和硫的变化轨迹。

6.2 变量逐步调整策略

由于汽油精制工业操作装置不可能实现无级调参，每次能够调整的操作变量仅仅是一个Δ值，既操作变量的单位操作值。

差分量子粒子群寻优算法能够稳定且快速的寻找到我们需要的条件，当然，前提是这个操作条件是汽油精制系统实际可达的。我们发现，如果要同时满足硫含量不大于 5μg/g，并且辛烷值损失降幅大于 30%的操作变量优化条件完全没有，但是我们找到了一个能够让目标函数损失最小的解，下面介绍这个最优解的逐步调整策略。

通过我们的策略，只需要 9 步就能够达到我们所需要的最优解。我们假设这一组变量中每一个变量不能瞬间跳跃两个及其以上的Δ值，但是这一组变量可以同时调节，每一个同时调节一个Δ值。那么，达到这组最优操作变量的最短步数取决于最优操作变量中需要操作次数最多的那个变量。

表 6-1 原操作变量与目标操作变量所需操作次数

变量名称	还原器流化氢气流量	催化汽油进装置总流量	原料进装置温度	原料换热器进口温度	D104 温度
原操作量	648.4958	122.7257	61.52	54.7885	124.868
目标操作量	798.4958	132.7257	66.52	62.7885	128.868
Δ	50	5	1	1	1
操作次数	3	2	5	8	4

变量名称	还原器温度	产品汽油出装置流量	S-ZORB.FT_1204.TOTAL	E-101 壳程出口总管温度	D-123 凝结水入流
原操作量	261.8852	135.9969	232247.51	126.5145	1990.011
目标操作量	264.8852	205.9969	302247.51	129.5145	2790.011
Δ	1	10	10000	1	200
操作次数	3	7	7	3	4

变量名称	C-201 下进料管温度	K-101B 右排气温度	K-101B 排气压力	K-101B 进气压力	D101 原料缓冲罐压力
原操作量	120.615	*49.1861*	*3.119*	*2.2379*	*0.3506*
目标操作量	129.615	43.1861	2.869	2.3879	1.3506
Δ	1	*-1*	*-0.05*	*0.05*	*0.5*
操作次数	9	6	5	3	*2*

表 6-1 是原操作变量与目标操作变量所需操作次数，Δ表示我们每次能够操作的最小值，从表中我们可以看出，我们所需要的最多的操作为 9 步，那么我们逐步调整到最优策略的最短操作步长为 9 步。

6.3 目标优化过程的可视化

我们调整操作变量的策略如下，

一、调整所有的十五个操作变量，调整两步，那么催化汽油进装置总流量、和 D101 原料缓冲罐压力就退出了优化过程，因为他们已经到达了我们的目标操作量。

二、调整剩下的十三个变量，调整一步，则还原器流化氢气流量、还原器温度、E-101 壳程出口总管温度、K-101B 进气压力这几个操作变量达到目标操作量，退出优化过程。

三、调整剩下的九个变量，调整一步，那么 D104 温度和 D-123 凝结水入流退出优化过程。

四、调整一步，则原料进装置温度和 K-101B 排气压力退出优化过程。

五、调整一步，则 K-101B 右排气温度退出优化过程

六、调整一步，则产品汽油出装置流量、S-ZORB.FT_1204.TOTAL 退出优化过程

七、调整一步，则原料换热器进口温度退出优化过程

八、调整一步，C-201 下进料管温度退出优化过程。

以上一共用了 9 步，就能够将所有原 133 号样本的操作变量全部调整到目标的操作变量。

图 6-1，图 6-2 分别是我们利用第四章中建立的模型所进行的目标操作变量优化而引起的硫含量和辛烷值的变化图，以下对两幅图进行分析。

通过对 133 号样本逐步进行操作变量调整，我们得到的辛烷值变化情况如图 6-1 所示，我们能够发现产品辛烷值在 0.2 内变化。由题目所知，“辛烷值每降低 1 个单位，相当于损失约 150 元/吨。以一个 100 万吨/年催化裂化汽油精制装置为例，若能降低 RON 损失 0.3 个单位，其经济效益将达到四千五百万元。”在这里，我们通过差分量子粒子群寻优算法得到的最优变量操作条件能够使该企业获得三千万元的经济效益。

通过对 133 号样本进行逐步优化调整主要操作变量，我们得到的硫含量变化情况如图 6-1 所示。可以看出，通过我们的操作变量调整策略，硫的含量在逐步减小。但是，我们发现硫的变化在 0.1 微克每克的水平内细微变化，起始值为 3.3 微克每克，优化结束之后为 3.19 微克每克，变化量仅为 0.14 微克。这样的情况说明我们寻找到的最优操作变量方案虽然能够轻微降低产品中硫的含量，但是对硫含量的影响已经很微小了。

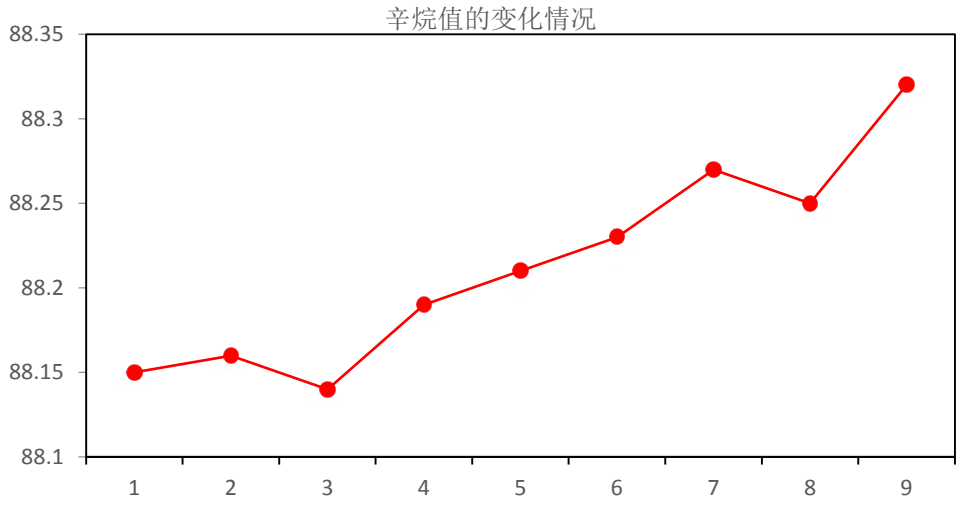


图 6-1 优化操作变量时辛烷值含量变化情况

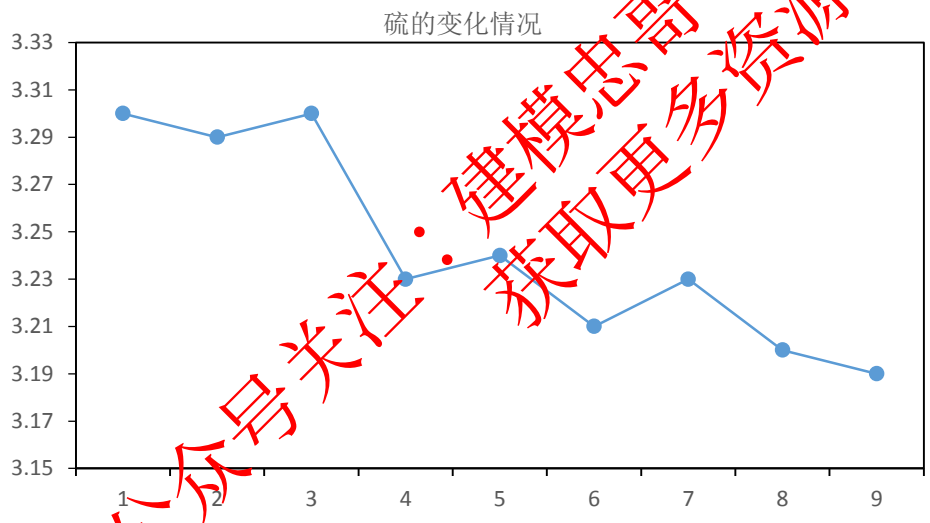


图 6-2 优化操作变量时硫含量变化情况

6.4 总结

（1）该脱硫保辛烷的汽油精制装置或者工艺在降低辛烷值损失的功能上还有潜力，如果能够利用我们提出的优化之后的操作变量，选取一个适当的调整策略，我们认为该装置还能够在降低辛烷损失值的功能上掘出更大的潜力；

（2）在降低产品硫含量的功能上已经快要到达了极限。如果需要进一步降低硫含量，我们建议更新老化生产设备、保障生产条件。这样才能够得到更大的环保效益和更高的经济效益。

第七章 模型评价

7.1 模型的优点

优点一：本文在构建模型时，采用了皮尔逊相关性分析，选取了与预测目标皮尔逊相关系数绝对值大于 0.4 的变量作为主要变量，同时通过 BP 神经网络进行综合分析，选取出 26 个主要变量。

优点二：通过 XGBoot、Light GBM、随机森林、支持向量机和 BP 神经网络的对产品中辛烷值的预测结果的对比，由 MAE、RMSE 和 MAPE 这三个评价指标，选取出预测精度最高的模型。该模型的精度高、收敛速度快并且鲁棒性强。

优点三：在操作变量的优化方案中，依次采用了粒子群算法、量子粒子群算法和差分量子粒子群算法对优化模型进行求解，筛选出辛烷值损失降幅大于等于30%的样本。通过不断对算法的改进，寻找到了更过的操作变量的优化方案。并且本文所设计的粒子群算法、量子粒子群算法和差分量子粒子群算法对其他问题同样适用。

7.2 模型的缺点

缺点一：因为产品的辛烷值损失数值较小，细微误差会体现为较大误差，因此对其直接进行预测会有较大的误差。

缺点二：操作变量的优化方案的优化效果和筛选样本效果都不够好，不能令人满意。这其中的原因不仅仅是算法自身的缺陷，也有可能是模型建立的不够科学或者 SVM 和 LGBM 的拟合精度还不够高。

参考文献

[1] 董立霞. FCC汽油加氢脱硫过程中烯烃饱和与辛烷值损失规律的研究[中国石油大学（北京），2017.

[2] Y. H, G. L, M. L, et al. Influenced factors and attainment rates of vapor recovery system for service stations in Beijing: 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet), 2011[C].2011
16-18 April 2011.

[3] Son-Ki I, Murid H. Carbon molecular sieves for catalyst supports: Thiophene hydrodesulfurization: 2008 3rd IEEE International Conference on Nano/Micro Engineered and Molecular Systems, 2008[C].2008
6-9 Jan. 2008.

[4] J. S, L. J. Octane Number Detection Based on Raman Spectra: 2010 International Conference on Electrical and Control Engineering, 2010[C].2010
25-27 June 2010.

[5] 黄涛，邓燕妮. 基于s-LTP和相似度匹配的人脸识别算法. 科技创新与应用, 2020(25):17-19.

[6] Curse of Dimensionality; Studies from E. Chavez et al Further Understanding of Curse of Dimensionality (Near neighbor searching with K nearest references). Computers, Networks & Communications, 2015.

[7] Study of Peak Load Demand Estimation Methodology by Pearson Correlation Analysis with Macro-economic Indices and Power Generation Considering Power Supply Interruption. Journal of Electrical Engineering & Technology, 2017,12(4).

[8] Zhao S, Zeng D, Wang W, et al. Mutation grey wolf elite PSO balanced XGBoost for radar emitter individual identification based on measured signals. Measurement, 2020,159.

[9] Dong W, Huang Y, Lehane B, et al. XGBoost algorithm-based prediction of concrete electrical resistivity for structural health monitoring. Automation in Construction, 2020,114.

[10] 오재영，함도현，이용진, et al. Short-term Load Forecasting Using XGBoost and the Analysis of Hyperparameters. 전기학회논문지, 2019,68(9).

[11] Kim J, Lee H, Oh J. Study on prediction of ship's power using light GBM and XGBoost. 한국마린엔지니어링학회지, 2020,44(2).

[12] Vasilev A, Sofi R, Rahman R, et al. Using Light for Therapy of Glioblastoma Multiforme (GBM). Brain Sciences, 2020,10(2).

[13] Sun P. Research on Credit Rating Model of P2P Project Based on Light GBM Algorithms: Research on Credit Rating Model of P2P Project Based on Light GBM Algorithms, 中国内蒙古呼和浩特, 2019[C].

[14] P. R D L, Fatichah C, Purwitasari D. Deteksi Gempa Berdasarkan Data Twitter Menggunakan Decision Tree, Random Forest, dan SVM. Jurnal Teknik ITS, 2017,6(1).

[15] Agajanian S, Oluyemi O, Verkhivker G M, et al. Integration of Random Forest Classifiers and Deep Convolutional Neural Networks for Classification and Biomolecular Modeling of Cancer Driver Mutations. Frontiers in Molecular Biosciences, 2019,6.

[16] Manessa M D M, Kanno A, Sekine M, et al. SATELLITE-DERIVED BATHYMETRY USING RANDOM FOREST ALGORITHM AND WORLDVIEW-2 IMAGERY. Geoplanning: Journal of Geomatics and Planning, 2016,3(2).

[17] D S A, Rajiv A. BP Components in Advanced CKD and the Competing Risks of Death, ESRD,

and Cardiovascular Events. Clinical journal of the American Society of Nephrology : CJASN, 2015,10(6).

[18] The BP Deepwater Horizon Disaster: Developing and Teaching a Business School Teaching Case as the Crisis Unfolded. Journal of Corporate Citizenship, 2016(61).

[19] García M M. Perception is truth: How U.S. newspapers framed the “Go Green” conflict between BP and Greenpeace. Public Relations Review, 2010,37(1).

[20] Z. Y, A. W, H. L, et al. An Elitist Promotion Quantum-Behaved Particle Swarm Optimization Algorithm: 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2016[C].2016
27-28 Aug. 2016.

[21] Y. F, M. D, C. Z, et al. Route Planning for Unmanned Aerial Vehicle (UAV) on the Sea Using Hybrid Differential Evolution and Quantum-Behaved Particle Swarm Optimization. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2013,43(6):1451-1465.

[22] şevklİ A Z, Sevİlgen F E. Discrete particle swarm optimization for the team orienteering problem. TURKISH JOURNAL OF ELECTRICAL ENGINEERING & COMPUTER SCIENCES, 2012,20(2).

[23] Li Y, Bai X, Jiao L, et al. Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. Applied Soft Computing, 2017,56.

[24] Leema N, Nehemiah H K, Kannan A. Quantum-Behaved Particle Swarm Optimization Based Radial Basis Function Network for Classification of Clinical Datasets. International Journal of Operations Research and Information Systems (IJORIS), 2018,9(2).