

基于决策树和多目标规划的信贷决策

摘要

中小微企业的规模相对较小，抵抗风险的能力不够，也缺少可抵押的固定资产，常常面临着贷款难、融资难的问题，因此研究中小微企业贷款策略对提高银行收益和促进中小微企业的发展具有重要的现实意义。

本文首先对已知数据进行了预处理，为模型建立提供数据基础。

对于问题一，本文从企业实力、供求关系稳定性和企业信誉方面入手建立**综合评价模型**对信贷风险进行评价量化，以上下游企业数量、2018-2020 年总营业额、2018-2019 营业额增长率、2018-2019 净利润增长率以及信誉等级等作为评价指标，建立评价体系，利用**熵权法**得到指标权重，再利用 **Topsis 法**对各个企业信贷风险进行评价，得到评价得分即为信贷风险的量化结果。

然后本文利用 **Logistic 回归**对信贷风险进行回归分析得到违约概率。依据银行的放贷原则，不对信誉评级为 D 的企业放贷，然后以违约概率、信贷风险以及银行政策等为约束，以银行利润最大和客户流失率最小为目标对剩余企业建立**多目标规划模型**，利用**主要目标法**进行求解得到信贷策略。当贷款总额为 6000 万元时，控制流失率小于 0.4 时得到的**最大利润为 361.65 万元**，控制收益大于 300 万元时优化流失率得到的**最小客户流失率为 36.29%**。

对于问题二，对无信贷记录的企业的信贷风险量化分析，本文利用附件一数据训练**决策树**，利用训练好的决策树对附件二数据进行预测分类，得到每个企业的违约情况和信誉等级，违约则信誉等级为 D。得到信誉等级后，利用问题一中的综合评价模型，得到量化后的信贷风险。

然后利用问题一的模型与方法将信贷风险转化为违约概率，并在其基础上，将年度信贷总额固定在 1 亿元，建立多目标规划模型进行优化，从而得到企业的信贷策略，此时控制流失率小于 0.4 时得到的**最大利润为 701.97 万元**，控制收益大于 600 万元时得到的**最小客户流失率为 35.77%**。

对于问题三，不同的突发情况会对不同的企业造成不同影响，本文对不同突发情况进行了讨论，选择对中小微企业影响较大的 2020 年新冠疫情影响和 2019 年浙江洪水自然灾害影响。然后根据突发因素期间的同比产值增长率作为影响程度，并重新对信贷风险进行量化分析，然后利用问题二中的模型，分别得到不同情况下的信贷策略，且**新冠疫情影响下控制流失率时得到的最大利润为 691.69 万元**，控制利润时得到的**最小客户流失率为 36.25%**；**洪水灾害影响下控制流失率时得到的最大利润为 693.33 万元**，控制利润时得到的**最小客户流失率为 35.79%**。最后我们将正常情况与突发情况下的最大收益和最小流失率进行比较，发现在突发因素的影响下，银行收益会相对降低而客户流失率会有所增长，且新冠疫情影响程度大于洪水灾害。

最后我们对本文中控制收益和流失率的界限值作了灵敏度分析，对模型进行了评价与改进，进一步优化了模型目标，最终对模型进行了应用与推广。

关键词：Logistic 回归；决策树；多目标规划；熵权法+Topsis

一、问题重述

1.1 问题的背景

随着我国经济的发展以及政策的完善，中小微企业在我国经济中占比逐渐增大，并在扩大就业、发展社会经济等方面发挥重大作用，但大部分中小微企业都面对着贷款难、融资难的问题，因此研究中小微企业贷款策略对提高银行利率和扶持和促进中小微企业的发展具有重要的现实意义。

现实中，中小微企业的规模相对较小，抵抗风险的能力不够，也缺少可抵押的固定资产，因此贷款银行往往是根据当前信贷政策、中小微企业信息以及上下游企业的影响力，向实力较强并且供求关系相对稳定的企业提供贷款；并为了扶持中小微企业发展，对信誉评级高、信贷风险较小的企业给予利率优惠。具体信贷流程如下：

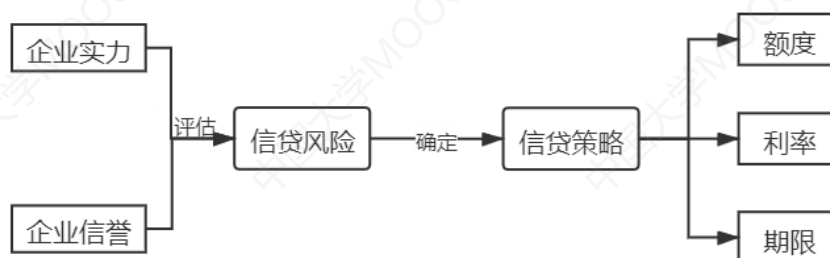


图 1-1 银行的信贷流程

1.2 问题的提出

某银行对确定要放贷企业的的贷款信息如下：贷款额度为10~100万元；年利率为4%~15%；贷款期限为1年。根据实际和附件 1~3 分别给出的 123 家有信贷记录企业的相关数据、302 家无信贷记录企业的相关数据和贷款利率与客户流失率关系的 2019 年统计数据，通过建立数学模型研究对中小微企业的信贷策略，主要解决下列问题：

问题一：（1）由相关数据对附件 1 中 123 家企业的信贷风险量化分析。

（2）若年度信贷总额固定，给出该银行对这些企业的信贷策略。

问题二：（1）在问题一的基础上，对附件 2 中 302 家企业的信贷风险进行量化分析。

（2）给出该银行在年度信贷总额为 1 亿元时对这些企业的信贷策略。

问题三：突发因素往往对不同行业、不同类别企业生产经营和经济效益有不同的影响。综合附件 2 中各企业的信贷风险和可能的突发因素（例如：新冠疫情）对各企业的影响，给出该银行年度信贷总额为 1 亿元时的信贷调整策略。

二、问题分析

首先我们对已知数据进行预处理，筛选出异常数据进行修正，并将数据中有参考价值的文本类数据进行量化，从而为模型建立提供数据基础。

针对问题一(1): 利用处理好的数据进行分析, 从企业实力、供求关系稳定性和企业信誉等方面入手, 得到上下游企业数量、2018-2020 年总营业额、2018-2019 营业额增长率、2018-2019 净利润增长率以及信誉等级等与企业信贷风险相关的因素, 并以这些因素作为评价指标建立评价体系, 利用综合评价模型对各个企业信贷风险进行评价, 得到评价得分即为量化结果。

针对问题一(2): 银行信贷策略的制定的标准往往为收益的大小以及客户流失情况, 由问题一(1)对信贷风险进行量化分析后可以由其使用 Logistic 回归得到违约概率, 而违约概率是计算贷款利润以及损失等数据的基本参数。我们依据银行不对信誉等级为 D 的企业贷款的原则, 然后利用违约概率、信贷风险以及银行政策等因素建立以利润最大和客户流失率最小为目标的多目标规划模型, 然后利用主要目标法进行求解得到信贷策略, 分别求解控制客户流失率优化收益情况下的信贷决策和控制收益优化客户流失率情况下的信贷决策。

针对问题二(1): 问题二是对无信贷记录的企业的信贷风险量化分析, 因此我们需要利用已知数据进行预测分类, 得到每个企业的违约情况和信誉等级, 违约情况则信誉等级为 D。得到信誉等级后, 重复问题一(1)中的综合评价方法, 得到量化后的信贷风险。

针对问题二(2): 将信贷风险转化为违约概率, 并在问题一的基础上, 将年度信贷总额固定在 1 亿元, 利用问题一(2)中的多目标规划模型对 302 家企业进行优化, 从而得到企业的信贷策略, 即贷款额度和利率。

针对问题三: 不同的突发情况会对不同的企业造成不同影响, 因此首先我们对不同突发情况进行讨论, 选择对中小微企业影响较大的新冠疫情影响和自然灾害影响。然后根据突发因素期间的经济指标变化作为影响信贷风险的因素, 重新对突发情况影响下的信贷风险进行量化评价, 然后代入问题 (2) 中的模型, 分别得到不同情况下的信贷策略。最后针对正常情况和突发情况下的银行收益和客户流失率进行比较, 从而看出突发因素的影响情况。

三、模型假设

- 1、假设突发情况对同一行业的影响是相同的;
- 2、假设银行贷款额度必须放完;
- 3、假设对于信誉等级为 D 的企业不给予贷款;

四、符号说明

符号	含义	单位
z_i	评价信贷风险的第 i 个评价指标($i=1,2,\dots,7$)	/
G_i	第 i 个企业的信贷风险评分($i=1,2,\dots,302$)	/
P_i	第 i 个企业的违约概率($i=1,2,\dots,302$)	/
l_j	第 j 项年利率对应的流失率($j=1,2,\dots,29$)	/

v_j	第 j 项年利率 ($j=1,2,\dots,29$)	/
z_1	银行所得总利润	万元
z_2	银行的总客户流失率	/

五、数据预处理

首先我们对附件 1~3 中的数据类型及数据内容进行分析，筛选出其中的异常数据进行修正，并将信用评级等文本类数据进行等级量化，得到对本文模型有参考价值的正确数据。

5.1 异常数据的修正

根据经济学知识并结合进项和销项发票信息可以得到：

$$\text{价税合计} = \text{金额} + \text{税额} \quad (1)$$

由式(1)对附件 1、2 中的发票信息进行检验共得到 5 个异常数据，并依据经济学知识对异常数据进行修正，结果如下：

表 5-1 异常数据的修正结果

企业代号	发票号码	异常数据类型	异常数据	修正后数据
E11	76244287	税额	0.68	-0.68
E13	37295688	税额	0.69	-0.69
E27	82240	税额	0.46	-0.46
E181	18415050	税额	0.32	-0.32
E181	18415168	税额	0.13	-0.13

由表 5-1 可以看出异常数据均为发票税额错误，且经修正后得到正确数据，有效保证了模型的准确性。

5.2 文本数据的量化

通过对数据类型的分析得到对本文有参考价值的文本类数据有是否违约、信用评级和发票状态三类数据，因此我们利用等级划分将这三类数据进行定量化处理，量化结果如下：

表 5-2 文本数据的等级量化

数据类型	数据内容	量化等级
是否违约	是	1
	否	0
信用评级	A	1
	B	2
	C	3

	D	4
发票状态	有效发票	1
	作废发票	0

六、模型的建立与求解

6.1 问题一的模型建立与求解

首先我们利用处理好的数据进行分析，得到与企业信贷风险相关的因素，并建立评价指标体系，利用综合评价模型对各个企业信贷风险进行量化分析。

然后利用定量化的信贷风险进行逻辑回归得到违约概率，结合违约概率等信息，以银行利润最大化和客户流失率最小为目标建立多目标规划，最终得到信贷策略。

6.1.1 基于综合评价的信贷风险量化分析

各个企业的信贷风险大小取决于**企业实力的强弱^[1]**、**供求关系是否稳定**以及**企业信誉的好坏**等因素，因此我们综合上述因素建立评价指标体系，利用综合评价的方法对 123 家企业的信贷风险进行量化分析，从而得到各个企业的信贷风险评分，具体分析流程如下：

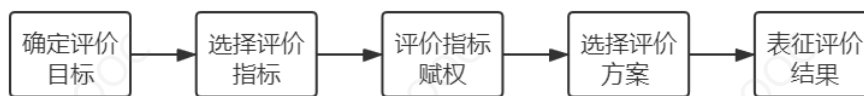


图 6-1 信贷风险评价的分析流程

(1) 评价指标体系的建立

通过对已知信息的筛选和处理，我们从企业供求关系、企业实力与企业信誉三方面考虑，并结合企业发展等因素，最终选取上下游企业数量、供求交易次数平均值、2018-2020 年总营业额等七个评价指标建立如下评价体系：

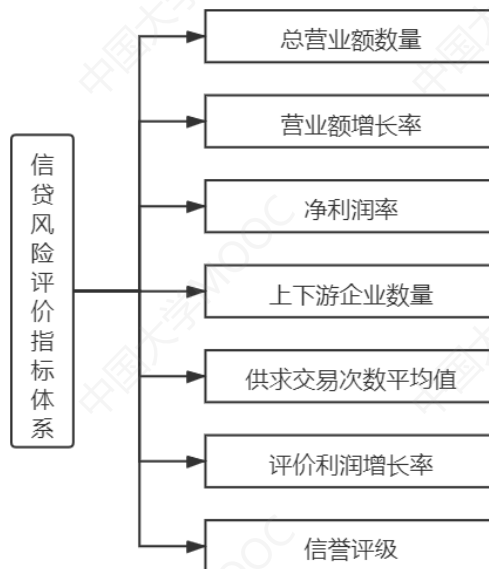


图 6-2 信贷风险的评价体系

结合企业信贷风险对评价指标进行分析如下：

1) 上下游企业数量(z_1):

上下游企业的数量即为企业进项与销项交易的企业数量，一定程度上反映了企业的供求情况，上下游企业数量越多，企业供求关系相对较稳定，企业风险相对较低。

2) 供求交易次数平均值(z_2):

企业与每家上下游企业的交易次数不同，所体现出的供求关系不同。因此我们取企业与其不同上下游企业交易次数的平均值，来体现该企业的供求关系稳定性，平均值越大，企业供求关系较稳定，信贷风险较低。

3) 2018-2020 年总营业额(z_3):

企业总营业额是企业实力的体现，这里我们取 2018-2020 年营业额为评价指标，总营业额越高，企业实力相对较强，信贷风险相对较低。

4) 2018-2019 年营业额增长率(z_4):

营业额增长率一定程度上体现了企业相对收入增长情况，反映了企业实力变化情况，营业额增长率越高，企业实力相对较强，信贷风险较低，其计算公式如下：

$$2019 \text{ 年营业额增长率} = \frac{2019 \text{ 年销项价税合计} - 2018 \text{ 年销项价税合计}}{2018 \text{ 年销项价税合计}} \quad (2)$$

5) 所有交易净利润率(z_5):

净利润率为一个企业的相对净收益的体现，反映了公司的整体实力，净利润率越高，公司实力越强，信贷风险越低，其计算公式如下：

$$\text{所有交易净利润率} = \frac{\text{销项总金额} - \text{进项总金额}}{\text{销项总金额}} \quad (3)$$

6) 2018-2019 年净利润增长率(z_6):

净利润增长率是企业经营趋势和稳定情况的体现，增长率越大则企业经营良好，实力逐渐增强，信贷风险也随之越低，其计算公式如下：

$$\text{净利润} = \text{销项总金额} - \text{进项总金额} \quad (4)$$

$$\text{2019 年净利润增长率} = \frac{\text{2019 年净利润} - \text{2018 年净利润}}{\text{2018 年净利润}} \quad (5)$$

7) 信誉评级(z_7):

信誉评级是根据企业贷款记录等情况对企业信誉作出的评价，我们从 1-4 对企业信誉评级 A-D 进行量化，故信誉评级越高，信贷风险越高。

将上述指标中的极小值变量转化为极大值变量，并基于以上 7 个评价指标的分析，我们需要对评价指标影响程度大小进行判别。而熵值可以判断多个指标对综合评价的影响（即权重）就越大，因此，我们可利用熵权法，计算出各个指标的权重，为多指标综合评价提供依据。

(2) 熵权法确定评价权重

熵值可以判断指标的离散程度，信息熵值越小，指标的离散程度越大，该指标的权重越大。具体计算步骤如下：

由于上述指标数据的数量级和量纲有差异，为了消除数量级与量纲的影响，首先我们利用下式对 123 家企业的数据和 7 个评价指标进行归一化处理得到标准化矩阵 Z :

$$Z_{ij} = \frac{z_{ij} - \max(z_{kj})}{\max(z_{kj}) - \min(z_{kj})} \quad (6)$$

其中 $i, k = 1, 2, \dots, 123$ 表示 123 家企业， $j = 1, 2, \dots, 7$ 分别表示 7 个评价指标。

计算第 j 项指标下第 i 个企业数据所占的比重 p_{ij} ($i = 1, 2, \dots, 123; j = 1, 2, \dots, 7$)，并将其看做相对熵计算中用到的概率，得到概率矩阵 P ，计算公式如下：

$$p_{ij} = \frac{Z_{ij}}{\sum_{i=1}^{123} Z_{ij}} \quad (j = 1, 2, \dots, 7) \quad (7)$$

计算第 j 个指标的信息熵 e_j ($j = 1, 2, \dots, 7$):

$$e_j = -\frac{1}{\ln 123} \sum_{i=1}^{123} p_{ij} \ln(p_{ij}) \quad (j = 1, 2, \dots, 7) \quad (8)$$

计算信息效用值 $d_j(j = 1, 2, \dots, 7)$ ，并归一化得到每个指标的熵权 $W_j(j = 1, 2, \dots, 7)$:

$$d_j = 1 - e_j \quad (j = 1, 2, \dots, 7) \quad (9)$$

$$W_j = \frac{d_j}{\sum_{j=1}^7 d_j} \quad (10)$$

由上述方法解得上下游企业数量(z_1)、供求交易次数平均值(z_2)、2018-2020 年总营业额(z_3)、2018-2019 营业额增长率(z_4)、所有交易净利润率(z_5)、2018-2019 净利润增长率(z_6)以及信誉等级(z_7)7 个指标的权重如下:

表 6-1 评价指标的权重系数表

指标	z_1	z_2	z_3	z_4	z_5	z_6	z_7
权重	0.2605	0.1002	0.4109	0.1514	0.0052	0.0021	0.0697

由表 6-1 可以看出 2018-2020 年总营业额对于信贷风险影响程度较大，2018-2019 净利润增长率影响程度较小。根据确定的权重，我们可以利用 Topsis 法对信贷风险进行综合评价。

(3) Topsis 法综合评价

Topsis 算法^[2]是一种常见的有限方案多目标的决策分析法。其基本原理为:计算各方案的综合评价值,然后根据综合评价值的大小对各评价对象进行排序。其方法步骤如下:

由 (2) 中对 123 次交易数据和 7 个评价指标进行正向化和标准化得到的标准化矩阵 Z 进行定义。

定义最大值 Z^+ :

$$Z^+ = (Z_1^+, Z_2^+, \dots, Z_j^+) = (\max\{z_{11}, z_{21}, \dots, z_{i1}\}, \dots, \max\{z_{1j}, z_{2j}, \dots, z_{ij}\}) \quad (11)$$

其中 $i = 1, 2, \dots, 123$ 表示 123 家企业; $j = 1, 2, \dots, 7$ 表示 7 个评价指标。

定义最小值 Z^- :

$$Z^- = (Z_1^-, Z_2^-, \dots, Z_j^-) = (\min\{z_{11}, z_{21}, \dots, z_{i1}\}, \dots, \min\{z_{1j}, z_{2j}, \dots, z_{ij}\}) \quad (12)$$

其中 $i = 1, 2, \dots, 123$ 表示 123 家企业; $j = 1, 2, \dots, 7$ 表示 7 个评价指标。

定义第 i ($i = 1, 2, \dots, 123$) 家企业与最大值的距离 D_i^+ :

$$D_i^+ = \sqrt{\sum_{j=1}^7 W_j * (Z_j^+ - z_{ij})^2} \quad (i = 1, 2, \dots, 123) \quad (13)$$

定义第 i ($i = 1, 2, \dots, 123$) 家企业与最小值的距离 D_i^- :

$$D_i^- = \sqrt{\sum_{j=1}^7 w_j * (Z_j^- - z_{ij})^2} \quad (i = 1, 2, \dots, 123) \quad (14)$$

那么我们可以计算出第 i 家企业的信贷风险评分 G_i :

$$G_i = \frac{D_i^-}{D_i^+ + D_i^-} \quad (i = 1, 2, \dots, 123) \quad (15)$$

由上述方法, 我们得到了 123 家企业的信贷风险评价得分频数图如下 (详细量化结果见附件):

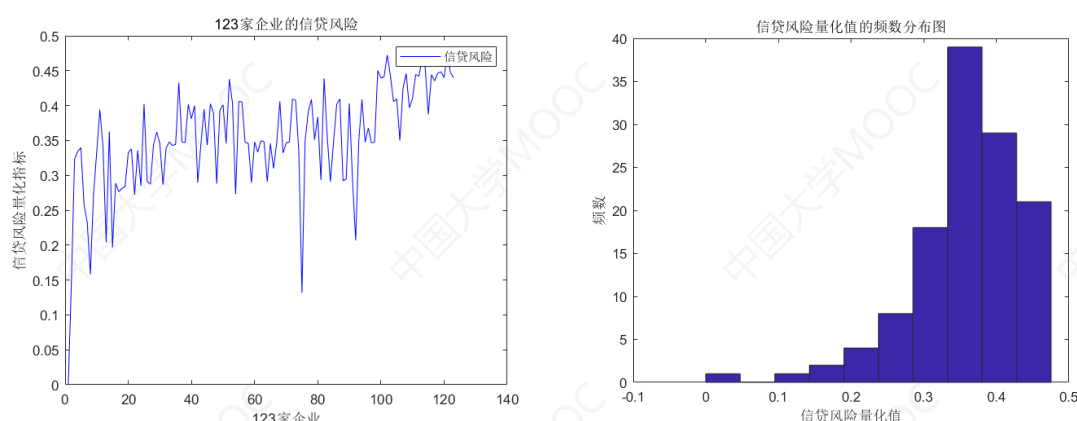


图 6-3 123 家企业的信贷风险评分频数分布图

由图 6-3 可以看出量化得分越高, 信贷风险越高且信贷风险量化得分主要集中在 0.3-0.5 范围内, 少数企业的信贷风险较低。

6.1.2 基于多目标规划的信贷策略模型

银行贷款定价需要考虑银行收益以及客户流失率两个目标, 而银行常常利用企业违约概率计算贷款预期损失、贷款定价以及信贷组合管理等数据。信贷风险在一定程度上反映了企业违约概率, 因此我们利用量化后的信贷风险求解得到企业违约概率。

企业违约概率为离散变量, Logistic 回归是数据挖掘中经常用到的分类算法, 它可以得到分类的概率, 因此我们采用 Logistic 回归求解违约概率。

(1) Logistic 回归求解违约概率

Logistic 回归是一种广义的线性回归模型, 设 x_1 为信贷风险, 得到 Logistic 回归方程如下:

$$\text{logit}(P) = \beta_0 + \beta_1 x_1 \quad (16)$$

可以记作:

$$\hat{y} = \text{logit}(P) = \ln \frac{p}{1-p} \quad (17)$$

由此得到违约概率 p 计算方程：

$$p = \frac{\exp(\beta_0 + \beta_1 x_1)}{1 + \exp(\beta_0 + \beta_1 x_1)} \quad (18)$$

利用 SPSS 软件对信贷风险进行 Logistic 逻辑回归，计算得到回归系数 $\beta_1=79.87$ ， $\beta_0=-32.96$ ，从而得到计算方程如下：

$$p = \frac{\exp(-32.96 + 79.87x_1)}{1 + \exp(-32.96 + 79.87x_1)} \quad (19)$$

由式（19）进行计算得到违约概率，并利用 SPSS 检验得到正确率为 95.7%，正确率较高，回归效果较好。

利用上述求得的违约概率和信誉等级等数据，我们可以建立多目标规划模型，实现银行的利润最大化和客户流失率最小。

(2) 多目标规划模型^[3]的建立

根据注（6）：银行原则上不给予信誉等级为 D 的企业进行贷款，因此我们不对 24 家信誉等级为 D 的企业放贷。

然后我们对附件 3 中 29 个年利率分别作为不同企业放贷的利率，对剩余 99 家企业（信誉等级 27 家为 A，38 家为 B，34 家为 C）建立多目标规划模型，确定放贷的额度和利率，期限为 1 年。

1) 决策变量：

① $x_i^{(k)}$ ：信誉等级为 k 的第 i 家企业($i = 1, 2, \dots, n^{(k)}$; $k = 1, 2, 3$)的放贷额度。

② $r_{ij}^{(k)} = \begin{cases} 0 & \text{信誉等级为 } k \text{ 的第 } i \text{ 家企业的利率不为第 } j \text{ 项年利率} \\ 1 & \text{信誉等级为 } k \text{ 的第 } i \text{ 家企业的利率为第 } j \text{ 项年利率} \end{cases} \quad (20)$

其中($i = 1, 2, \dots, n^{(k)}$; $j = 1, 2, \dots, 29$; $k = 1, 2, 3$)

2) 目标函数：

① 利润最大：

$$\max z_1 = \sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} \left[(1 - P_i^{(k)}) x_i^{(k)} \cdot \sum_{j=1}^{29} (r_{ij}^{(k)} v_j) - P_i^{(k)} x_i^{(k)} \right] \quad (21)$$

其中 $P_i^{(k)}$ 为信誉等级为 k 的第 i 个企业的违约概率； v_j 为第 j 项年利率。

② 客户流失率最小：

$$\min z_2 = \frac{\sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} \sum_{j=1}^{29} (r_{ij}^{(k)} l_j)}{N} \quad (22)$$

其中 l_j 为第 j 个年利率对应的客户流失率。

3) 约束条件:

① 信贷总额约束: 保证放贷总额为固定值 a 。

$$\sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} x_i^{(k)} = a \quad (23)$$

② 额度范围约束: 保证每个信誉等级的公司对应相应的额度范围且不超过总体范围。

$$90 \leq x_i^{(1)} \leq 100 \quad (24)$$

$$70 + 10 \cdot \frac{0.01 - P_i^{(2)}}{|P_i^{(2)} - 0.01|} \leq x_i^{(2)} \leq 85 + 10 \cdot \frac{0.01 - P_i^{(2)}}{|P_i^{(2)} - 0.01|} \quad (25)$$

$$15 + 5 \cdot \frac{0.05 - P_i^{(3)}}{|P_i^{(3)} - 0.05|} \leq x_i^{(3)} \leq 25 + 5 \cdot \frac{0.05 - P_i^{(3)}}{|P_i^{(3)} - 0.05|} \quad (26)$$

其中 $i=1,2,\dots,n^{(k)}$; 式(25)表达的意思是当 $P_i^{(2)} \geq 0.01$ 时, $x_i^{(2)}$ 的取值固定在[60,75]内, 否则固定在[80,95]内, 式(26)同理。

③ 利率存在约束: 保证每个企业都会有相应的利率。

$$\sum_{j=1}^{29} r_{ij}^{(k)} = 1 \quad (i = 1, 2, \dots, n^{(k)}) \quad (k = 1, 2, 3) \quad (27)$$

④ 利率范围约束: 保证每个信誉等级的企业对应相应的利率取值范围。

$$r_{ij}^{(1)} = 0 \quad (i = 1, 2, \dots, n^{(1)}) \quad j \geq 7 \quad (28)$$

$$r_{ij}^{(2)} = 0 \quad (i = 1, 2, \dots, n^{(2)}) \quad j \leq 7, j \geq 15 \quad (29)$$

$$r_{ij}^{(3)} = 0 \quad (i = 1, 2, \dots, n^{(3)}) \quad j \leq 15, j \geq 21 \quad (30)$$

综上所述, 对 99 家企业建立的多目标规划模型如下:

$$\max z_1 = \sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} \left[(1 - P_i^{(k)}) x_i^{(k)} \cdot \sum_{j=1}^{29} (r_{ij}^{(k)} v_j) - P_i^{(k)} x_i^{(k)} \right]$$

$$\min z_2 = \frac{\sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} \sum_{j=1}^{29} (r_{ij}^{(k)} l_j)}{N}$$

$$s. t. \begin{cases} \sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} x_i^{(k)} = a \\ 90 \leq x_i^{(1)} \leq 100 \\ 70 + 10 \cdot \frac{0.01 - P_i^{(2)}}{|P_i^{(2)} - 0.01|} \leq x_i^{(2)} \leq 85 + 10 \cdot \frac{0.01 - P_i^{(2)}}{|P_i^{(2)} - 0.01|} \\ 15 + 5 \cdot \frac{0.05 - P_i^{(3)}}{|P_i^{(3)} - 0.05|} \leq x_i^{(3)} \leq 25 + 5 \cdot \frac{0.05 - P_i^{(3)}}{|P_i^{(3)} - 0.01|} \\ \sum_{j=1}^{29} r_{ij}^{(k)} = 1 \\ r_{ij}^{(1)} = 0 \quad j \geq 7 \\ r_{ij}^{(2)} = 0 \quad j \leq 7, j \geq 15 \\ r_{ij}^{(3)} = 0 \quad j \leq 15, j \geq 21 \end{cases} \quad (31)$$

其中 $n^{(1)}=27$, $n^{(2)}=38$, $n^{(3)}=24$ 。

(3) 主要目标法求解多目标规划模型

这里我们用主要目标法对模型进行求解，即将目标作为主要目标，其余目标作为次要目标，确定适当的界限值，把次要目标作为约束，将多目标优化问题转化为求解单目标规划问题，这里我们令 $a=6000$ ，利用 Lingo 对模型进行求解。

1) 控制客户流失率优化收益的单目标规划模型

首先考虑银行收益最大，将客户流失率最小作为次要目标，设定界限值得到如下约束：

$$\frac{\sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} \sum_{j=1}^{29} (r_{ij}^{(k)} l_j)}{N} \leq 0.4 \quad (j = 1, 2, \dots, 29) \quad (32)$$

将客户流失率约束添加到上述规划模型中，求解得到部分信贷策略如下（详细策略见附件）：

表 6-2 部分企业信贷策略

企业代号	贷款额度（万元）	贷款利率
E1	90	0.4
E2	90	0.4
⋮	⋮	⋮
E122	0	0
E123	0	0

此时的最大总收益为 361.65 万元。

2) 控制收益优化客户流失率的单目标规划模型

考虑客户流失率最小为主要目标时，银行收益最大为次要目标，设定界限值得到银行收益约束如下：

$$\sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} \left[(1 - P_i^{(k)}) x_i^{(k)} \cdot \sum_{j=1}^{29} (r_{ij}^{(k)} v_j) - P_i^{(k)} x_i^{(k)} \right] > 300 \quad (33)$$

将客户流失率约束添加到上述规划模型中，利用 Lingo 求解得到信贷策略如下：

表 6-3 部分企业信贷策略

企业代号	贷款额度	贷款利率
E1	90.07271	0.04
E2	90.0727	0.04
⋮	⋮	⋮
E122	0	0
E123	0	0

此时最小平均客户流失率为 36.29%。

6.2 问题二的模型建立与求解

首先对于问题二的信贷风险量化分析，由于该 302 家企业无信誉等级数据，需要利用企业信息及发票数据对 302 家企业信誉等级进行预测分类，再利用问题一中的评价指标体系对信贷风险进行量化分析。

得到量化分析的信贷风险后，利用问题一中的 Logistic 回归模型计算出违约概率，并在年度信贷总额为 1 亿元时建立多目标规划模型，从而得到 302 家企业具体信贷策略。

6.2.1 基于决策树的信誉等级分类

决策树又称分类回归树，分为分类树和回归树。其中的分类树通常用于对离散变量的分类预测，具有较高的准确性，且经常应用于经济金融领域，因此我们使用分类树对是否违约情况进行分类，再由各项指标对信誉等级分类。

(1) 分类指标的选取

对是否违约分类需要选取公司实力、供求关系稳定性以及发票作废率等反映公司经营情况的指标。对企业信息进行统计分析我们得到相关指标有**是否违约、总净利率、上下游企业数量、供求关系稳定性、2018-2020 年总营业额、2018-2019 年营业额增长率、2018-2019 年净利润增长率以及无效和负数发票数量** 8 个分类指标。

利用决策树分类时若指标之间存在较高的相关性，则会产生过拟合的情况，影响分类结果的精确度，因此我们首先对 8 个指标进行相关性分析：

表 6-4 分类指标的相关性矩阵

	是否违约	总净利率	上下游企业个数	供求关系稳定性	总营业额	营业额增长率	净利润增长率	无效和负数发票数
是否违约	1.0000	-0.2571	-0.1915	-0.1604	-0.1240	0.0151	0.0632	-0.1937
总净利率	-0.2571	1.0000	0.0486	0.0613	-0.0032	0.0593	-0.0153	0.0588
购方销售方个数	-0.1915	0.0486	1.0000	-0.0746	0.1500	-0.0943	0.0149	0.5585
供求关系稳定性	-0.1604	0.0613	-0.0746	1.0000	0.1614	-0.0604	0.0518	0.2388
总营业额	-0.1240	-0.0032	0.1500	0.1614	1.0000	-0.0693	0.0007	0.2774
营业额增长率	0.0151	0.0593	-0.0943	-0.0604	-0.0693	1.0000	-0.7102	-0.0971
净利润增长率	0.0632	-0.0153	0.0149	0.0518	0.0007	-0.7102	1.0000	0.0220
无效和负数发票数	-0.1937	0.0588	0.5585	0.2388	0.2774	-0.0971	0.0220	1.0000

由表 6-4 可以看出，各个指标之间均没有较大的相关性，故可选用该 8 个指标进行分类。

(2) 建立决策树模型

1) 决策树的原理

决策树^[4]的原理是将已知数据从根节点（树的最高层节点）排列到某个叶子节点来进行分类，决策树上的非叶节点表示对属性取值的测试，其分支代表测试结果；而树上的一个叶子节点代表一个分类的类别，当所有叶子节点给出的分类结果都一样时，就结束生长，即已经可以判定样本的类别。

2) 决策树的构建

这里我们将信息增益作为节点选择的度量，即采用 ID3 算法构造决策树。

设 S 为样本数据集合，并将 S 中元素分为 n 类，那么 S 相对于 n 个类别的分类熵定义为：

$$Entropy(S) = \sum_{i=1}^n -p_i \log_2 p_i \quad (34)$$

其中 p_i 为 S 中第 i 个分类的样本数所占比例，且熵值越小，子集划分程度越高。

由熵值我们可以得到信息增益 $Gain(S, A)$ ：

$$Gain(S, A) = Entropy(S) - \sum_{v \in V(A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad (35)$$

其中 $V(A)$ 为属性 A 的值域； S_v 为 S 在属性 A 的划分下值为 v 的子集。

基于上述定义，我们得到基于信息增益为根节点选择的决策树构建流程如下：

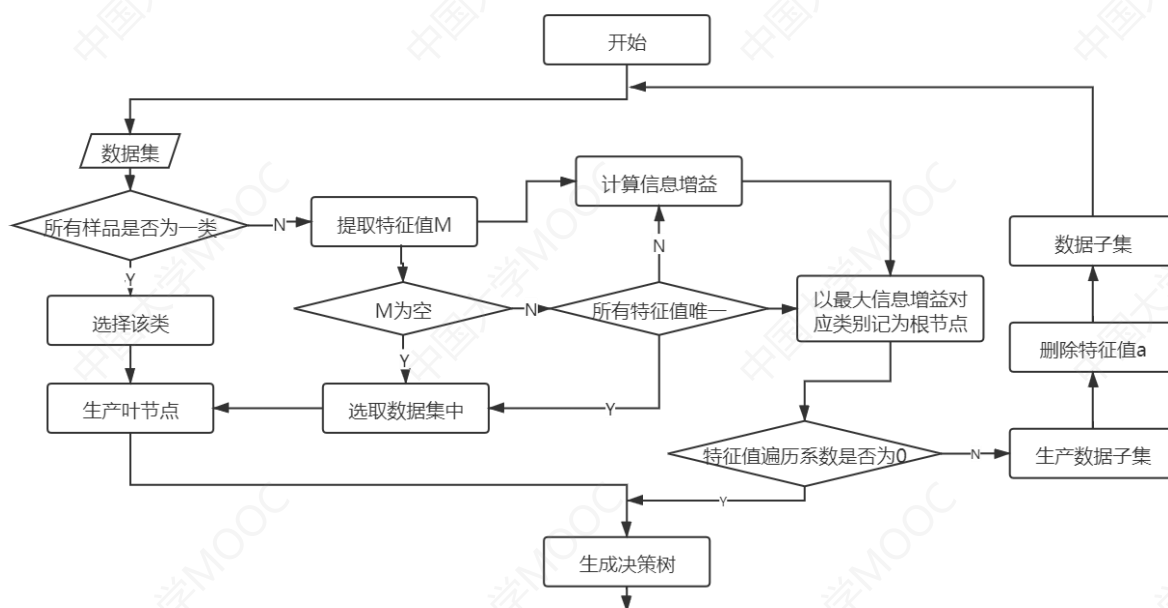


图 6-4 决策树的构建流程^[5]

3) 决策树的剪枝

为了防止决策树分支过多，导致过拟合的情况出现，我们需要对决策树进行后剪枝。即在决策树构建完成后对树上非叶节点进行分析，判断将该节点替换为叶节点是否会带来泛化性能的提升，即验证集精度的提高，若可以则将其替换为叶节点。

(3) 决策树模型的求解

基于上述算法分析，我们首先由选取的 8 个指标对企业是否违约进行多次预测分类，然后取每个企业多次分类中概率最高的分类结果，最终得到结果为 **41 家企业违约**，并将其信誉等级分为 **D**，在测试集上的**测试准确性为 80.1%**，分类结果可以接受。

再利用决策树模型对未违约企业的信誉等级进行分类，分类**准确率为 71.1%**，结果不满意，因此我们对其进行多次分类，最后取众数得到比较准确的分类结果，并分别将其分为 A、B、C 三类，**分类结果为：31 家企业信誉等级为 A，209 家企业信誉等级为 B，21 家企业为信誉等级 C**（详细分类结果见附件）。

6.2.2 基于综合评价的信贷风险量化分析

通过上述方法对企业信誉等级的分类，进一步完善了综合评价指标数据，依据问题一中综合评价方法对 302 家企业信贷风险进行评价。

首先利用熵权法对评价指标赋权，得到上下游企业数量(z_1)、供求交易次数平均值(z_2)、2018-2020 年总营业额(z_3)、2018-2019 营业额增长率(z_4)、所有交易净利润率(z_5)、2018-2019 净利润增长率(z_6)以及信誉等级(z_7)**7 个指标的权重如下：**

表 6-5 评价指标的权重系数表

指标	z_1	z_2	z_3	z_4	z_5	z_6	z_7
权重	0.2256	0.1287	0.3287	0.2652	0.0010	0.0010	0.0497

由表 6-5 可以看出对于该 302 家企业的信贷风险，2018-2020 年总营业额对于信贷风险影响程度较大，所有交易净利润率和 2018-2019 净利润增长率影响程度较小。

根据确定的权重，我们再利用 Topsis 法对信贷风险进行综合评价，得到信贷风险评价得分图如下(具体评分见附件):

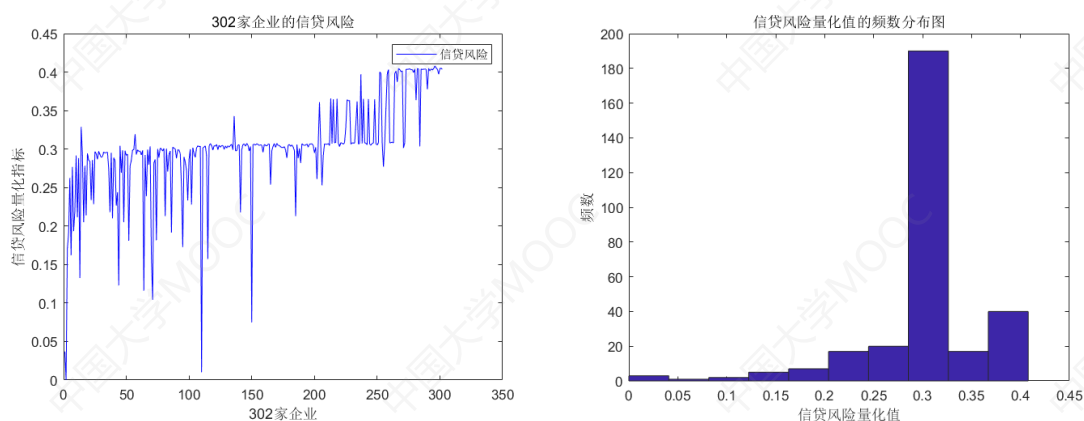


图 6-5 302 家企业的信贷风险评分及分布图

由图 6-5 可以分析得到，302 家企业的信贷风险量化主要集中在 0.3 左右，有个别企业信贷风险较小。

6.2.3 基于多目标规划的信贷策略模型

(1) Logistic 逻辑回归求解违约概率

针对问题二中的信贷策略，我们依据问题一中的方法，首先利用信贷风险对违约概率进行 Logistic 逻辑回归，利用 SPSS 计算得到回归系数 $\beta_1=120.254$ ， $\beta_0=-53.904$ ，故计算方程如下：

$$p = \frac{\exp(-53.904 + 120.254x_1)}{1 + \exp(-53.904 + 120.254x_1)} \quad (36)$$

由 SPSS 检验得到正确率为 99.3%，正确率较高，回归效果较好。

(2) 多目标规划模型的建立与求解

由 6.2.1 对违约信誉等级的分类，我们得到有 41 家信誉等级为 D 的企业并不予放贷，利用问题一中的模型对剩余 261 家（信誉等级 31 家为 A，209 家为 B，21 家为 C）企业建立多目标规划模型如下：

$$\max z_1 = \sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} \left[(1 - P_i^{(k)}) x_i^{(k)} \cdot \sum_{j=1}^{29} (r_{ij}^{(k)} v_j) - P_i^{(k)} x_i^{(k)} \right]$$

$$\min z_2 = \frac{\sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} \sum_{j=1}^{29} (r_{ij}^{(k)} l_j)}{N}$$

$$\text{s. t. } \begin{cases} \sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} x_i^{(k)} = a \\ 90 \leq x_i^{(1)} \leq 100 \\ 70 + 10 \cdot \frac{0.01 - P_i^{(2)}}{|P_i^{(2)} - 0.01|} \leq x_i^{(2)} \leq 85 + 10 \cdot \frac{0.01 - P_i^{(2)}}{|P_i^{(2)} - 0.01|} \\ 15 + 5 \cdot \frac{0.05 - P_i^{(3)}}{|P_i^{(3)} - 0.05|} \leq x_i^{(3)} \leq 25 + 5 \cdot \frac{0.05 - P_i^{(3)}}{|P_i^{(3)} - 0.05|} \\ \sum_{j=1}^{29} r_{ij}^{(k)} = 1 \\ r_{ij}^{(1)} = 0 \quad j \geq 7 \\ r_{ij}^{(2)} = 0 \quad j \leq 7, j \geq 15 \\ r_{ij}^{(3)} = 0 \quad j \leq 15, j \geq 21 \end{cases} \quad (37)$$

其中 $n^{(1)}=31$, $n^{(2)}=209$, $n^{(3)}=21$ 。

利用主要目标法对上述模型进行求解得到如下结果：

1) 银行收益最大为主要目标

客户流失率最小作为次要目标，设定界限值得到如下约束：

$$\frac{\left[\sum_{i=1}^{27} r_{ij}^{(A)} l_j + \sum_{i=1}^{38} r_{ij}^{(B)} l_j + \sum_{i=1}^{34} r_{ij}^{(C)} l_j \right]}{261} < 0.4 \quad (j = 1, 2, \dots, 29) \quad (38)$$

将客户流失率约束添加到上述规划模型中，利用 Lingo 求解得到信贷策略如下：

表 6-6 部分企业信贷策略

企业代号	贷款额度	贷款利率
E124	80	0.0585
E125	80	0.0585
⋮	⋮	⋮
E301	0	0
E302	0	0

此时最大总收益为 701.97 万元。

2) 客户流失率最小为主要目标

银行收益最大为次要目标，设定界限值得到银行收益约束如下：

$$\sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} \left[(1 - P_i^{(k)}) x_i^{(k)} \cdot \sum_{j=1}^{29} (r_{ij}^{(k)} v_j) - P_i^{(k)} x_i^{(k)} \right] > 600 \quad (39)$$

将客户流失率约束添加到上述规划模型中，利用 Lingo 求解得到信贷策略如下：

表 6-7 部分企业信贷策略

企业代号	贷款额度	贷款利率
E124	80	0.04
E125	80	0.04
⋮	⋮	⋮
E301	0	0
E302	0	0

此时最小评价客户流失率为 35.77%。

6.3 问题三的模型建立与求解

当发生新冠疫情、自然灾害以及美国经济制裁等突发事件时，不同行业企业的生产经营和经济效益会随之受到不同影响，而此时的信贷策略也应发生改变。

表 6-8 突发事件的影响区域及行业

突发事件	影响区域	影响行业
新冠病毒疫情	全球	部分行业收益、部分行业受损
洪水、干旱等自然灾害	部分城市	大部分行业受损
美国经济制裁	全国	高科技行业受损

分析表 6-8，针对突发事件对中小微企业的影响，这里我们以 2020 年新冠病毒疫情和 2019 年浙江洪水灾害为例，量化分析突发事件对不同行业、不同类别的企业带来的影响，从而给出相应的信贷策略。

6.3.1 企业的分类

首先我们根据企业的经营内容及规模将 302 家企业大致分为个体经营类、服务类、制造类、建筑安装类、批发类、物流类、科技类、贸易类、网络科技类、广告文化类、园林类、农业类、医疗卫生类、餐饮零售类以及房地产类 15 种企业，例如下表所示：

表 6-9 企业分类与代表企业

代表企业	分类	代表企业	分类
个体经营 E124	个体经营户	***食品有限公司	批发类

***工程咨询有限公司	服务类	***运类有限公司	物流类
***建设工程有公司	建筑安装类	***电子科技有限公司	科技类
***机械有限责任公司	制造类	***工贸有限公司	贸易类
***网络信息安全有限公司	网络科技类	***广告传媒股份有限公司	广告文化类
***生态魔芋有限公司	农业类	***医疗有限公司	医疗卫生类
***建电管理咨询有限公司	通信类	***林园艺场	园林类
***房地产营销策划有限公司	房地产类		

6.3.2 突发事件影响下的信贷策略

基于问题二的模型分析，我们首先将突发事件期间各个行业的影响程度进行量化，然后将量化后的数据纳入问题二中的评价指标体系，从而推出突发情况下各个企业的信贷风险，再根据问题二的方法建立多目标规划模型，得到突发事件影响下的信贷策略。

(1) 新冠疫情影响程度的量化^[6]

疫情对不同行业所带来的影响不同，例如在疫情的影响下长期停工的建筑业、餐饮业等行业严重受损；而网络科技业、医疗器械制造业等行业会因疫情需求而受益。因此我们需要对不同种类企业的影响程度进行量化，这里我们选取疫情期间同比去年的产值增长率作为影响程度的量化指标对企业进行量化（详细量化结果见附件）：

表 6-10 新冠疫情对不同种类企业的影响程度

企业代号	企业分类	影响程度
E124	个体经营类	-34.4%
E125	个体经营类	-34.4%
⋮	⋮	⋮
E141	批发类	16.97%
⋮	⋮	⋮
E301	贸易类	-6.8%
E302	贸易类	-6.8%

(2) 洪水灾害影响程度的量化

洪水对除医疗卫生类的所有行业所带来的影响基本均为负面影响，而农业类等依赖天气的行业则受损更为严重。我们以 2019 年浙江洪水灾害为例，这里我们同样选取同比去年的产值增长率作为影响程度的量化指标对企业进行量化（详细量化结果见附件）：

表 6-11 洪水灾害对不同种类企业的影响程度

企业代号	企业分类	影响程度
E124	个体经营类	-4.5%

E125	个体经营类	-4.5%
⋮	⋮	⋮
E301	贸易类	-2%
E302	贸易类	-2%

(3) 基于多目标规划的信贷策略

突发事件的影响会造成公司实力的变化，从而影响信贷风险。因此得到量化后的影响程度后，将其作为评价指标，利用问题二中的评价模型和 Logistic 逻辑回归求得信贷风险和违约概率，此时逻辑回归的正确率为 95.7% 和 98%，回归情况较好。

得到违约概率后，代入如下模型：

$$\max z_1 = \sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} \left[(1 - P_i^{(k)}) x_i^{(k)} \cdot \sum_{j=1}^{29} (r_{ij}^{(k)} v_j) - P_i^{(k)} x_i^{(k)} \right]$$

$$\min z_2 = \frac{\sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} \sum_{j=1}^{29} (r_{ij}^{(k)} l_j)}{N}$$

$$s. t. \begin{cases} \sum_{k=1}^3 \sum_{i=1}^{n^{(k)}} x_i^{(k)} = 10000 \\ 90 \leq x_i^{(1)} \leq 100 \\ 70 + 10 \cdot \frac{0.01 - P_i^{(2)}}{|P_i^{(2)} - 0.01|} \leq x_i^{(2)} \leq 85 + 10 \cdot \frac{0.01 - P_i^{(2)}}{|P_i^{(2)} - 0.01|} \\ 15 + 5 \cdot \frac{0.05 - P_i^{(3)}}{|P_i^{(3)} - 0.05|} \leq x_i^{(3)} \leq 25 + 5 \cdot \frac{0.05 - P_i^{(3)}}{|P_i^{(3)} - 0.05|} \\ \sum_{j=1}^{29} r_{ij}^{(k)} = 1 \\ r_{ij}^{(1)} = 0 \quad j \geq 7 \\ r_{ij}^{(2)} = 0 \quad j \leq 7, j \geq 15 \\ r_{ij}^{(3)} = 0 \quad j \leq 15, j \geq 21 \end{cases} \quad (40)$$

其中 $n^{(1)}=31$, $n^{(2)}=209$, $n^{(3)}=21$ 。

对上述模型进行求解得到部分疫情影响和洪水灾害影响下的企业信贷策略如下（详细策略见附件）：

表 6-12 疫情影响下部分企业信贷策略

企业代号	以收益最大为主要目标		以客户流失率最小为主要目标	
	贷款额度	贷款利率	贷款额度	贷款利率
E124	80	0.04	80	0.04
E125	80	0.04	80	0.04
⋮	⋮	⋮	⋮	⋮
E301	0	0	0	0
E302	0	0	0	0

疫情影响下控制流失率时得到的最大利润为 691.69 万元，控制利润时得到的最小客户流失率为 36.25%。

表 6-13 洪水灾害影响下部分企业信贷策略

企业代号	以收益最大为主要目标		以客户流失率最小为主要目标	
	贷款额度	贷款利率	贷款额度	贷款利率
E124	80	80	80	80
E125	80	80	80	80
⋮	⋮	⋮	⋮	⋮
E301	0	0	0	0
E302	0	0	0	0

洪水灾害影响下控制流失率时得到的最大利润为 693.33 万元，控制利润时得到的最小客户流失率为 35.79%。

(4) 正常情况与突发因素影响下信贷策略对比

我们以问题二中得到的信贷策略作为正常情况下的信贷策略，并与问题三中两种突发情况下的信贷策略进行比较，得到突发因素对收益与客户流失率的影响，得到相比正常情况下收益和流失率变化率：

$$\text{收益变化率} = \frac{Z_1 \text{ 突发} - Z_1 \text{ 正常}}{Z_1 \text{ 正常}} \quad (41)$$

$$\text{流失率变化率} = \frac{Z_2 \text{ 突发} - Z_2 \text{ 正常}}{Z_2 \text{ 正常}} \quad (42)$$

由式(41)和(42)得到下列比较结果：

表 6-14 控制流失率时最大收益比较

情况	最大收益（万元）	相比正常情况变化率
正常情况	701.97	/
新冠疫情影响	691.64	-1.97%

洪水灾害影响	693.33	-1.18%
--------	--------	--------

表 6-15 控制收益时最小流失率比较

情况	最小流失率（万元）	相比正常情况变化率
正常情况	35.77%	/
新冠疫情影响	36.25%	1.34%
洪水灾害影响	35.79%	0.08%

由表 6-14 和表 6-15 分析得到，在突发因素的影响下，收益会相对正常情况降低，而客户流失率会相对增高，且洪水灾害影响小于新冠疫情影响。

七、灵敏度分析

7.1 问题一中控制客户流失率的灵敏度分析

在问题一优化模型的求解中，控制客户流失率优化收益。现在对该控制流失率进行调整，得到不同控制流失率时的最优收益。

表 7-1 不同控制流失率时的最大收益

控制流失率	0.44	0.43	0.42	0.41	0.40	0.39	0.38	0.37
最大收益（万元）	388.31	388.31	379.49	370.84	361.65	353.05	344.19	335.54

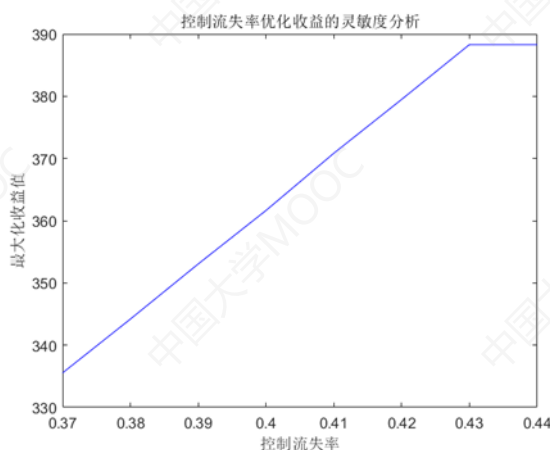


图 7-1 控制流失率优化收益的灵敏度分析

根据上述图表可知随着控制流失率的降低，最大利润在不断的降低，是符合实际的一种情况。

7.2 问题一中控制收益的灵敏度分析

在问题一优化模型的求解中，控制收益优化客户流失率。现在对该控制收益进行调整，得到不同控制收益时的最优客户流失率。

表 7-2 不同控制收益时的最小客户流失率

控制收益 (万元)	290	310	330	350	370	390	410	430
最小流失率	0.363	0.363	0.365	0.387	0.409	0.432	0.456	0.495

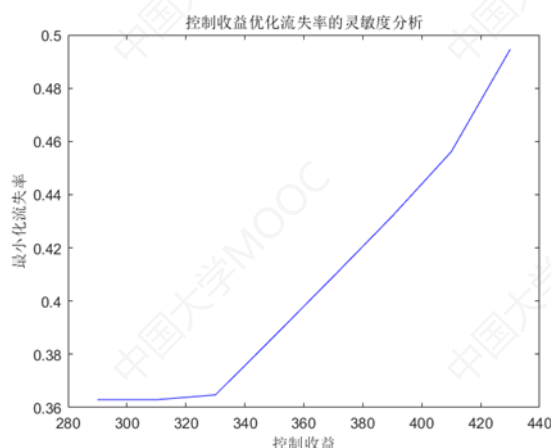


图 7-2 控制收益优化流失率的灵敏度分析

根据上述图表可知随着控制收益的增加，最小流失率在上升，且在控制收益小的时候客户流失率上升较缓，因此可以考虑将收益控制在此范围内，可以一定程度上减少客户流失率。

八、模型的评价与改进

8.1 模型的评价

8.1.1 模型的优点

- 1、本文首先进行了数据预处理和基本统计分析，修正了异常值，并得到了数据的大体分布，对模型建立提供了数据基础。
- 2、本文利用了决策树、Logistic 逻辑回归等常用的分类方法，使模型中变量的确定更加科学合理。
- 3、本文建立的多目标规划模型引入了 0-1 变量，很好地将年利率进行离散处理，使模型更加准确简洁。

8.1.2 模型的缺点

- 1、本文对数据的挖掘不够充分，原数据中可能存在很多数据进行挖掘。

- 2、本文在选择变量过程中，没有很好的考虑到各因素之间的交互作用，可能使结果产生一定误差。
- 3、本文建立的优化模型中某些参数的确定具有一定的主观性。
- 4、本文在使用决策树对信誉等级进行分类时，分类准确性还有待提升，可以考虑使用集成算法，如随机森林、AdaBoost 和 XGBoost 来提高准确性。

8.2 模型的改进

本文建立了多目标规划模型，是在控制次要目标作为约束，对主要目标进行求解，使收益最大化或流失率最小，但是在现实生活中，往往需要对收益 z_1 和流失率 z_2 进行综合分析，因此我们引入偏好系数 ρ 对目标函数进行改进：

$$\max z = \rho z_1 - (1 - \rho)z_2 \quad (43)$$

由于收益和流失率量纲不同，我们需要对其标准化处理，得到目标函数如下：

$$\max = \rho \frac{z_1 - \min \{z_1\}}{\max \{z_1\} - \min \{z_1\}} - (1 - \rho) \frac{z_2 - \min \{z_2\}}{\max \{z_2\} - \min \{z_2\}} \quad (44)$$

其中， $0 \leq \rho \leq 1$ 是一个合适的常数。

通过对偏好系数的修改，可以得到不同收益和流失率的综合情况，从而得到想要的优化目标。

九、模型的应用与推广

本文利用决策树、Logistic 回归等方法对对象进行分类，然后建立多目标优化模型给出不同情况下的信贷策略。该模型可应用于经济金融行业对风险的评估及各种定价的策略给出，并对各行业中小微企业的融资贷款具有参考价值，具有一定的科学性和适用性。

十、参考文献

- [1] <https://www.jianshu.com/p/1193b148dd7e>
- [2] 韩中庚.数学建模方法及其应用[M].第三版.北京：高等教育出版社，2017.12
- [3] <https://blog.csdn.net/Zengmeng1998/article/details/105724974>
- [4] <http://www.lishimeiyi.cn/fun/163.html>
- [5] 赵静娴.基于决策树的信用风险评估方法研究[D].天津：天津大学,管理学院,2009.5
- [6] https://www.sohu.com/a/372198342_120209965

附录 1:

自己查阅的数据（问题三）

新馆疫情对企业影响	自然灾害影响(洪水、台风)
-34.4	-4.5
-34.4	-4.5
-34.4	-4.5
-34.4	-4.5
-34.4	-4.5
-34.4	-4.5
-34.4	-4.5
-34.4	-4.5
-34.4	-4.5
-13	-4
-17.6	-3
-10.2	-5
-17.6	-3
-34.4	-4.5
-34.4	-4.5
-17.6	-3
16.97	-5
-26.2	-7
3.8	-0.5
-13	-4
-6.8	-2
-17.6	-3
-17.6	-3
-6.8	-2
-17.6	-3
-17.6	-3
-17.6	-3
-6.8	-2
-34.4	-4.5
-13	-4
-34.4	-4.5
-34.4	-4.5
-10.2	-5
-17.6	-3

-34.4	-4.5
-10.2	-5
-17.6	-3
13.2	-0.5
-26.2	-7
-34.4	-4.5
5	-0.1
-17.6	-3
5	-0.1
-10.2	-5
-17.6	-3
-17.6	-3
-13	-4
-26.2	-7
-6.8	-2
3.8	-0.5
16.97	-5
3.8	-0.5
13.2	-0.5
-6.8	-2
-5.1	-2
13.2	-0.5
16.97	-5
-10.2	-5
-17.6	-3
-17.6	-3
-3.4	-12.8
-17.6	-3
-34.4	-4.5
-10.2	-5
-13	-4
-13	-4
-6.8	-2
-17.6	-3
-34.4	-4.5
5	-0.1
72	2
5	-0.1
72	2
-6.8	-2

-17.6	-3
-34.4	-4.5
-26.2	-7
-34.4	-4.5
-17.6	-3
3.8	-0.5
-34.4	-4.5
-17.6	-3
-34.4	-4.5
-34.4	-4.5
-13	-4
-17.6	-3
-34.4	-4.5
13.2	-0.5
3.8	-0.5
-17.6	-3
-17.6	-3
-17.6	-3
-34.4	-4.5
-17.6	-3
-13	-4
5	-0.1
-13	-4
3.8	-0.5
3.8	-0.5
-4.2	-5
-13	-4
-17.6	-3
-13	-4
5	-0.1
-6.8	-2
-17.6	-3
-17.6	-3
-13	-4
-13	-4
-10.2	-5
-34.4	-4.5
-34.4	-4.5
-34.4	-4.5
-34.4	-4.5

-34.4	-4.5
-34.4	-4.5
-34.4	-4.5
-34.4	-4.5
3.8	-0.5
-34.4	-4.5
-17.6	-3
-13	-4
-10.2	-5
3.8	-0.5
-17.6	-3
-10.2	-5
72	2
5	-0.1
-10.2	-5
-17.6	-3
-34.4	-4.5
-26.2	-7
-17.6	-3
-26.2	-7
5	-0.1
5	-0.1
72	2
-34.4	-4.5
13.2	-0.5
-34.4	-4.5
-10.2	-5
-10.2	-5
-5.1	-2
-17.6	-3
-13	-4
-34.4	-4.5
3.8	-0.5
-34.4	-4.5
-34.4	-4.5
-17.6	-3
13.2	-0.5
-10.2	-5
-10.2	-5
-6.8	-2

-10.2	-5
-34.4	-4.5
-13	-4
5	-0.1
-10.2	-5
-10.2	-5
-34.4	-4.5
5	-0.1
-4.2	-5
-34.4	-4.5
-26.2	-7
-10.2	-5
5	-0.1
-26.2	-7
-26.2	-7
-34.4	-4.5
-10.2	-5
-17.6	-3
-17.6	-3
-13	-4
-10.2	-5
-17.6	-3
-13	-4
-10.2	-5
5	-0.1
3.8	-0.5
-13	-4
-34.4	-4.5
13.2	-0.5
-44.3	-3
-34.4	-4.5
-13	-4
-34.4	-4.5
-4.2	-5
-26.2	-7
-16.3	-0.7
-10.2	-5
-34.4	-4.5
3.8	-0.5
-6.8	-2

-34.4	-4.5
-10.2	-5
-26.2	-7
-10.2	-5
-17.6	-3
-6.8	-2
-10.2	-5
-10.2	-5
-34.4	-4.5
3.8	-0.5
-5.1	-2
-44.3	-3
5	-0.1
-6.8	-2
-16.3	-0.7
3.8	-0.5
-13	-4
-3.4	-12.8
-34.4	-4.5
-6.8	-2
-4.2	-5
-26.2	-7
-4.2	-5
-10.2	-5
-44.3	-3
-10.2	-5
3.8	-0.5
-34.4	-4.5
-10.2	-5
-44.3	-3
5	-0.1
5	-0.1
5	-0.1
-17.6	-3
-13	-4
5	-0.1
-44.3	-3
-17.6	-3
-44.3	-3
-44.3	-3

-44.3	-3
-17.6	-3
-13	-4
-10.2	-5
-17.6	-3
3.8	-0.5
3.8	-0.5
-16.3	-0.7
3.8	-0.5
13.2	-0.5
13.2	-0.5
-13	-4
3.8	-0.5
-13	-4
-34.4	-4.5
-44.3	-3
-44.3	-3
-13	-4
5	-0.1
-6.8	-2
72	2
-10.2	-5
13.2	-0.5
-13	-4
-44.3	-3
-34.4	-4.5
-17.6	-3
-34.4	-4.5
-3.4	-12.8
-6.8	-2
-13	-4
-10.2	-5
-44.3	-3
3.8	-0.5
-6.8	-2
-13	-4
13.2	-0.5
5	-0.1
-44.3	-3
72	2

-17.6	-3
-4.2	-5
-34.4	-4.5
-6.8	-2
-10.2	-5
-34.4	-4.5
-13	-4
-13	-4
-17.6	-3
-10.2	-5
-10.2	-5
13.2	-0.5
-10.2	-5
-6.8	-2
-34.4	-4.5
-26.2	-7
5	-0.1
3.8	-0.5
-5.1	-2
-13	-4
3.8	-0.5
72	2
-10.2	-5
-44.3	-3
-10.2	-5
-6.8	-2
-6.8	-2

附录 2:

本文源程序

本文代码使用的软件是 MatlabR2019b 和 Lingo11

问题一代码:

EntropyMethod.m 熵权法

my_log.m 自己定义的 log 函数

problem1_2.m 附件 1 数据处理代码

problem1_3.m 附件 1 信贷风险量化代码

problem1_4.m 附件 1 数据处理代码

问题 1.lg4 问题一优化模型求解

问题二代码:

problem2_1.m 附件 2 数据处理代码
 problem2_2.m 附件 2 决策树分类是否违约代码
 problem2_3.m 附件 2 决策数对于 A,B,C 等级分类代码
 problem2_4.m 附件 2 信贷风险量化代码
 problem2_5.m 附件 2 数据处理代码
 问题 2.lg4 问题二优化模型求解
 问题三代码:
 problem3_1.m 问题三量化信贷风险和数据处理代码 1(新冠疫情)
 problem3_2.m 问题三量化信贷风险和数据处理代码 2(自然灾害)
 问题 3.1.lg4 问题三一优化模型求解
 问题 3.2.lg4 问题三二优化模型求解
 数据导出代码:
 output_result.m 整理结果数据代码
 灵敏度分析 diamagnetic:
 range_analysis.m

说明:由于有些代码重复性很高,附录只给出一组代码,源代码详情见支撑材料。

Problem1_2.m problem1_4.m 数据处理代码
<pre> 1. clc 2. clear all; 3. 4. tic 5. sheet1 = xlsread('附件 1_2.xlsx',1,'C2:D124'); %企业信息 6. sheet2_1 = xlsread('附件 1_2.xlsx',2,'A2:A210948'); %进项发票 7. sheet2_2 = xlsread('附件 1_2.xlsx',2,'D2:G210948'); 8. sheet2_3 = xlsread('附件 1_2.xlsx',2,'I2:I210948'); 9. %第一列企业代号 第二列销方单位代号 第三列金额 第四列税额 第五列价税合计 第六列发票状态 1 有效 0 作废 10. sheet2 = [sheet2_1 sheet2_2 sheet2_3]; %进项发票矩阵 11. n1 = size(sheet2,1); 12. %统计负数发票 1 为有效 -1 为负数发票 13. negative_1 = ones(n1,1); 14. negative_1(sheet2(:,5)<0) = -1; 15. sheet2 = [sheet2 negative_1]; 16. sheet3_1 = xlsread('附件 1_2.xlsx',3,'A2:A162485'); 17. sheet3_2 = xlsread('附件 1_2.xlsx',3,'D2:G162485'); 18. sheet3_3 = xlsread('附件 1_2.xlsx',3,'H2:H162485'); 19. sheet3 = [sheet3_1 sheet3_2 sheet3_3]; 20. n2 = size(sheet3,1); 21. negative_2 = ones(n2,1); </pre>

```

22. negative_2(sheet3(:,5)<0) = -1;
23. %销项发票信息
24. sheet3 = [sheet3 negative_2];
25.
26. %看 sheet2 中交易+税额=价税合计
27. sum((sheet2(:,3)+sheet2(:,4)-sheet2(:,5))<=0.0001);
28. error1 = []; %2 个异常数据
29. for i = 1:n1
30.     if (sheet2(i,3)+sheet2(i,4)-sheet2(i,5))>=0.0001
31.         error1 = [error1;i];
32.     end
33. end
34. sum((sheet3(:,3)+sheet3(:,4)-sheet3(:,5))<=0.0001);
35. error2 = []; %1 个异常数据
36. for i = 1:n2
37.     if (sheet3(i,3)+sheet3(i,4)-sheet3(i,5))>=0.0001
38.         error2 = [error2;i];
39.     end
40. end
41.
42. in_2018_1 = xlsread('附件 1_2.xlsx',4,'A2:A79845');
43. in_2018_2 = xlsread('附件 1_2.xlsx',4,'D2:H79845');
44. negative = ones(size(in_2018_1,1),1);
45. negative(in_2018_2(:,4)<0) = -1; %负数发票改为-1
46. in_2018 = [in_2018_1 in_2018_2 negative]; %2018 年进项发票信息
47. in_2019_1 = xlsread('附件 1_2.xlsx',5,'A2:A92453');
48. in_2019_2 = xlsread('附件 1_2.xlsx',5,'D2:H92453');
49. negative = ones(size(in_2019_1,1),1);
50. negative(in_2019_2(:,4)<0) = -1;
51. in_2019 = [in_2019_1 in_2019_2 negative]; %2019 年进项发票信息
52. out_2018_1 = xlsread('附件 1_2.xlsx',6,'A2:A55537');
53. out_2018_2 = xlsread('附件 1_2.xlsx',6,'D2:H55537');
54. negative = ones(size(out_2018_1,1),1);
55. negative(out_2018_2(:,4)<0) = -1;
56. out_2018 = [out_2018_1 out_2018_2 negative]; %2018 年销项发票信息
57. out_2019_1 = xlsread('附件 1_2.xlsx',7,'A2:A53771');
58. out_2019_2 = xlsread('附件 1_2.xlsx',7,'D2:H53771');
59. negative = ones(size(out_2019_2,1),1);
60. negative(out_2019_2(:,4)<0) = -1;
61. out_2019 = [out_2019_1 out_2019_2 negative]; %2019 年销项发票信息
62. toc

```

```

63.
64. company_num = size(sheet1,1); %企业数量
65. credit = sheet1(:,2); %信誉等级
66. %将 123 家企业数据放入元胞数组中
67. cell_in = cell(company_num,1); %进项
68. cell_out = cell(company_num,1); %销项
69. index1 = [1];
70. index2 = [1];
71. for i = 1:n1-1
72.     if sheet2(i+1,1) ~= sheet2(i,1)
73.         index1 = [index1;i+1];
74.     end
75. end
76.
77. for i = 1:n2-1
78.     if sheet3(i+1,1) ~= sheet3(i,1)
79.         index2 = [index2;i+1];
80.     end
81. end
82.
83. for i = 1:1:company_num
84.     if i == 1
85.         cell_in{i,1} = sheet2(1:index1(2)-1,:);
86.         cell_out{i,1} = sheet3(1:index2(2)-1,:);
87.     elseif i == company_num
88.         cell_in{i,1} = sheet2(index1(end):end,:);
89.         cell_out{i,1} = sheet3(index2(end):end,:);
90.     else
91.         cell_in{i,1} = sheet2(index1(i):index1(i+1)-1,:);
92.         cell_out{i,1} = sheet3(index2(i):index2(i+1)-1,:);
93.     end
94. end
95.
96. %计算相关变量
97. %计算净利润（不含税）
98. cost = zeros(company_num,1); %成本
99. money = zeros(company_num,1); %营业额
100. for i = 1:company_num
101.     matrix1 = cell2mat(cell_in(i,1));
102.     matrix2 = cell2mat(cell_out(i,1));
103.     cost(i) = sum(matrix1(:,3));

```

```

104.     money(i) = sum(matrix2(:,3));
105. end
106. profit = money-cost; %全部发票的净利润率
107. profit_rate = profit./money; %净利润率
108. %供求关系稳定性指标
109. %计算代号个数和单个代号的平均值
110. %一、不同销方和购方的数量
111. numbers = zeros(company_num,1);
112. for i = 1:company_num
113.     set1 = []; %销方
114.     set2 = []; %购方
115.     matrix1 = cell2mat(cell_in(i,1));
116.     matrix2 = cell2mat(cell_out(i,1));
117.     for j = 1:size(matrix1,1)
118.         if sum(set1==matrix1(j,2)) ~= 1 %没有相同的销方
119.             set1 = [set1;matrix1(j,2)];
120.         end
121.     end
122.     for j = 1:size(matrix2,1)
123.         if sum(set2==matrix2(j,2)) ~= 1
124.             set2 = [set2;matrix2(j,2)];
125.         end
126.     end
127.     numbers(i) = length(set1)+length(set2);
128. end
129.
130. %二、单个代号重复次数的均值
131. fre_mean = zeros(company_num,1);
132. for i = 1:company_num
133.     max_id1 = max(sheet2(:,2));
134.     max_id2 = max(sheet3(:,2));
135.     max_id = max([max_id1 max_id2]);
136.     id_vec = zeros(max_id,1); %记录购方出现了几次
137.     matrix1 = cell2mat(cell_in(i,1));
138.     matrix2 = cell2mat(cell_out(i,1));
139.     for j = 1:size(matrix1,1)
140.         if matrix1(j,6)==1 && matrix1(j,7)==1 %只计算有效发票的数量
141.             id_vec(matrix1(j,2)) = id_vec(matrix1(j,2)) + 1;
142.         end
143.     end
144.     for j = 1:size(matrix2,1)

```

```

145.         if matrix2(j,6)==1 && matrix2(j,7)==1
146.             id_vec(matrix2(j,2)) = id_vec(matrix2(j,2)) + 1;
147.         end
148.     end
149.     %总次数
150.     all_num = sum(id_vec);
151.     %购方和销方个数
152.     plat_num = sum(id_vec~=0);
153.     fre_mean(i) = all_num/plat_num;
154. end
155.
156. %2018 年和 2019 年总营业额 out_2018 out_2019
157. cell_2018_out = cell(company_num,1);
158. cell_2019_out = cell(company_num,1);
159. %分组
160. index1 = [1];
161. n1 = size(out_2018,1);
162. n2 = size(out_2019,1);
163. for i = 1:n1-1
164.     if out_2018(i+1,1) ~= out_2018(i,1)
165.         index1 = [index1;i+1];
166.     end
167. end
168.
169. index2 = zeros(company_num,1);
170. index2(1,1) = 1;
171. for i = 1:n2-1
172.     if out_2019(i+1,1) ~= out_2019(i,1)
173.         index2(out_2019(i+1,1)) = i+1;
174.     end
175. end
176.
177. %2019 年销项缺少 112 117 121 企业的信息
178. for i = 1:company_num
179.     if i == 1
180.         cell_2018_out{i,1} = out_2018(1:index1(2)-1,:);
181.     elseif i == company_num
182.         cell_2018_out{i,1} = out_2018(index1(end):end,:);
183.     else
184.         cell_2018_out{i,1} = out_2018(index1(i):index1(i+1)-1,:);
185.     end

```

```

186. end
187.
188. for i = 1:company_num
189.     if i == 112 || i == 117 || i == 121
190.         continue
191.     elseif i == 1
192.         cell_2019_out{i,1} = out_2019(1:index2(2)-1,:);
193.     elseif i == company_num
194.         cell_2019_out{i,1} = out_2019(index2(end):end,:);
195.     elseif i == 112-1
196.         cell_2019_out{i,1} = out_2019(index2(i):index2(i+2)-1,:);
197.     elseif i == 117-1
198.         cell_2019_out{i,1} = out_2019(index2(i):index2(i+2)-1,:);
199.     elseif i == 121-1
200.         cell_2019_out{i,1} = out_2019(index2(i):index2(i+2)-1,:);
201.     else
202.         cell_2019_out{i,1} = out_2019(index2(i):index2(i+1)-1,:);
203.     end
204. end
205.
206. %2018 年和 2019 年进项数据
207. cell_2018_in = cell(company_num,1);
208. cell_2019_in = cell(company_num,1);
209. %分组
210. index1 = zeros(company_num,1);
211. index1(1,1) = 1;
212. index2 = zeros(company_num,1);
213. index2(1,1) = 1;
214. n1 = size(in_2018,1);
215. n2 = size(in_2019,1);
216. for i = 1:n1-1
217.     if in_2018(i+1,1) ~= in_2018(i,1)
218.         index1(in_2018(i+1,1)) = i+1;
219.     end
220. end
221.
222. for i = 1:n2-1
223.     if in_2019(i+1,1) ~= in_2019(i,1)
224.         index2(in_2019(i+1,1)) = i+1;
225.     end
226. end

```

```

227.
228. %2018 年进项缺少 42 94 104 数据
229. %2019 年进项缺少 69 115 117 数据
230. for i = 1:company_num
231.     if i == 42 || i == 94 || i == 104
232.         continue;
233.     elseif i == 1
234.         cell_2018_in{i,1} = in_2018(1:index1(2)-1,:);
235.     elseif i == company_num
236.         cell_2018_in{i,1} = in_2018(index1(end):end,:);
237.     elseif i == 42-1
238.         cell_2018_in{i,1} = in_2018(index1(i):index1(i+2)-1,:);
239.     elseif i == 94-1
240.         cell_2018_in{i,1} = in_2018(index1(i):index1(i+2)-1,:);
241.     elseif i == 104-1
242.         cell_2018_in{i,1} = in_2018(index1(i):index1(i+2)-1,:);
243.     else
244.         cell_2018_in{i,1} = in_2018(index1(i):index1(i+1)-1,:);
245.     end
246. end
247.
248. for i = 1:company_num
249.     if i == 69 || i == 115 || i == 117
250.         continue;
251.     elseif i == 1
252.         cell_2019_in{i,1} = in_2019(1:index2(2)-1,:);
253.     elseif i == company_num
254.         cell_2019_in{i,1} = in_2019(index2(end):end,:);
255.     elseif i == 69-1
256.         cell_2019_in{i,1} = in_2019(index2(i):index2(i+2)-1,:);
257.     elseif i == 115-1
258.         cell_2019_in{i,1} = in_2019(index2(i):index2(i+2)-1,:);
259.     elseif i == 117-1
260.         cell_2019_in{i,1} = in_2019(index2(i):index2(i+2)-1,:);
261.     else
262.         cell_2019_in{i,1} = in_2019(index2(i):index2(i+1)-1,:);
263.     end
264. end
265.
266. %计算 2018 和 2019 负数和无效发票数之和
267. negative_num1 = zeros(company_num,1);

```

```

268. negative_num2 = zeros(company_num,1);
269. for i = 1:company_num
270.     matrix1 = cell2mat(cell_2018_in(i,1));
271.     matrix2 = cell2mat(cell_2019_in(i,1));
272.     matrix3 = cell2mat(cell_2018_out(i,1));
273.     matrix4 = cell2mat(cell_2019_out(i,1));
274.     %2018
275.     if sum(matrix1)==0 & sum(matrix3)==0
276.         negative_num1(i) = 0;
277.     elseif sum(matrix1)==0
278.         negative_num1(i) = size(matrix3,1)-
            sum(matrix3(:,6)==1 & matrix3(:,7)==1);
279.     elseif sum(matrix3)==0
280.         negative_num1(i) = size(matrix1,1)-
            sum(matrix1(:,6)==1 & matrix1(:,7)==1);
281.     else
282.         negative_num1(i) = size(matrix3,1)-
            sum(matrix3(:,6)==1 & matrix3(:,7)==1)+size(matrix1,1)-
            sum(matrix1(:,6)==1 & matrix1(:,7)==1);
283.     end
284.     %2019
285.     if sum(matrix2)==0 & sum(matrix4)==0
286.         negative_num2(i) = 0;
287.     elseif sum(matrix2)==0
288.         negative_num2(i) = size(matrix4,1)-
            sum(matrix4(:,6)==1 & matrix4(:,7)==1);
289.     elseif sum(matrix4)==0
290.         negative_num2(i) = size(matrix2,1)-
            sum(matrix2(:,6)==1 & matrix2(:,7)==1);
291.     else
292.         negative_num2(i) = size(matrix4,1)-
            sum(matrix4(:,6)==1 & matrix4(:,7)==1)+size(matrix2,1)-
            sum(matrix2(:,6)==1 & matrix2(:,7)==1);
293.     end
294. end
295. negative_num = negative_num1+negative_num2;
296.
297. % negative_num1 = zeros(company_num,1); %2018
298. % negative_num2 = zeros(company_num,1); %2019
299. % for i = 1:company_num
300. %     matrix1 = cell2mat(cell_2018_in(i,1));

```



```

301. %     matrix2 = cell2mat(cell_2019_in(i,1));
302. %     matrix3 = cell2mat(cell_2018_out(i,1));
303. %     matrix4 = cell2mat(cell_2019_out(i,1));
304. %     if sum(matrix1)==0 & sum(matrix3)==0
305. %         negative_num1(i) = 0;
306. %     elseif sum(matrix1)==0
307. %         negative_num1(i) = sum(matrix3(:,6)==0 | matrix3(:,7)==-1);
308. %     elseif sum(matrix3)==0
309. %         negative_num1(i) = sum(matrix1(:,6)==0 | matrix1(:,7)==-1);
310. %     else
311. %         negative_num1(i) = sum(matrix3(:,6)==0 | matrix3(:,7)==-
1)+sum(matrix1(:,6)==0 | matrix1(:,7)==-1);
312. %     end
313. %     %2019
314. %     if sum(matrix2)==0 & sum(matrix4)==0
315. %         negative_num2(i)=0;
316. %     elseif sum(matrix2)==0
317. %         negative_num2(i) = sum(matrix4(:,6)==0 | matrix4(:,7)==-1);
318. %     elseif sum(matrix4)==0
319. %         negative_num2(i) = sum(matrix2(:,6)==0 | matrix2(:,7)==-1);
320. %     else
321. %         negative_num2(i) = sum(matrix4(:,6)==0 | matrix4(:,7)==-
1)sum(matrix2(:,6)==0 | matrix2(:,7)==-1);
322. %     end
323. % end
324.
325. %计算 2018 和 2019 年总营业额
326. %计算 2018-2019 营业额增长率
327. money = zeros(company_num,1);
328. money_rate = zeros(company_num,1);
329. for i = 1:company_num
330.     matrix1 = cell2mat(cell_2018_out(i,1));
331.     matrix2 = cell2mat(cell_2019_out(i,1));
332.     if sum(matrix1)==0
333.         money_rate(i) = 0;
334.         money(i) = sum(matrix2(:,5));
335.     elseif sum(matrix2) == 0
336.         money_rate(i) = 1;
337.         money(i) = sum(matrix1(:,5));
338.     else
339.         money(i) = sum(matrix1(:,5))+sum(matrix2(:,5));

```

```

340.         money_rate(i) = (sum(matrix2(:,5))-
        sum(matrix1(:,5)))/sum(matrix1(:,5));
341.     end
342. end
343.
344. %计算 2018-2019 净利润增长
345. net_rate = zeros(company_num,1);
346. net_2019 = zeros(company_num,1);
347. net_2018 = zeros(company_num,1);
348. for i = 1:company_num
349.     matrix1 = cell2mat(cell_2018_in(i,1));
350.     matrix2 = cell2mat(cell_2019_in(i,1));
351.     matrix3 = cell2mat(cell_2018_out(i,1));
352.     matrix4 = cell2mat(cell_2019_out(i,1));
353.     if sum(matrix4)==0 & sum(matrix2)==0
354.         net_2019(i) = 0;
355.     elseif sum(matrix4) == 0
356.         net_2019(i) = 0-sum(matrix2(:,3));
357.     elseif sum(matrix2) == 0
358.         net_2019(i) = sum(matrix4(:,3));
359.     else
360.         net_2019(i) = sum(matrix4(:,3))-sum(matrix2(:,3));
361.     end
362.
363.     if sum(matrix3) == 0
364.         net_2018(i) = 0-sum(matrix1(:,3));
365.     elseif sum(matrix1) == 0
366.         net_2018(i) = sum(matrix3(:,3));
367.     else
368.         net_2018(i) = sum(matrix3(:,3))-sum(matrix1(:,3));
369.     end
370. end
371. net_rate = (net_2019-net_2018)./net_2018;
372.
373. clc
374. clear all;
375.
376. risk = xlsread('result.xlsx',2,'A2:A124'); %信贷风险 越大风险越高
377. plot(risk)
378. normplot(risk)
379. [h,p] = jbtest(risk,0.05);

```

```

380.
381. result = xlsread('result.xlsx',3,'F2:G124');
382. %删除信誉等级为D的企业
383. ret = result(result(:,1)~=4,:);
384. retA = ret(ret(:,1)==1,:);
385. retB = ret(ret(:,1)==2,:);
386. retC = ret(ret(:,1)==3,:);

```

Problem1_3.m EntropyMethod.m my_log.m 信贷风险量化代码

```

1. clc
2. clear all;
3.
4. data = xlsread('问题一指标.xlsx',1,'A2:I124');
5. data(:,3) = [];
6. %第一列企业代号 第二列信誉等级 第三列全部发票净利润率 第四列不同代号个数
7. %第五列单个代号重复次数均值 第六列 2018-2019 总营业额 第七列 2018-2019 营业额增长率
   第八列 2018-2019 净利润增长率
8. %Min-Max 标准化
9. n = size(data,1);
10. data(:,1) = [];
11. std_data = (data-repmat(min(data),n,1))./(repmat(max(data),n,1)-
    repmat(min(data),n,1));
12. [R] = corrcoef(std_data);
13.
14. %信誉等级极小型指标
15. data(:,1) = max(data(:,1))-data(:,1);
16. std_data = (data-repmat(min(data),n,1))./(repmat(max(data),n,1)-
    repmat(min(data),n,1));
17. weight = EntropyMethod(std_data);
18. D_P = sum([(repmat(max(std_data),n,1)-
    std_data).^2].*repmat(weight,n,1),2).^0.5;
19. D_N = sum([(repmat(min(std_data),n,1)-
    std_data).^2].*repmat(weight,n,1),2).^0.5;
20. score = D_N./(D_P+D_N); %分数越高风险越小
21. risk = max(score)-score; %调整为正向 分数越高风险越大
22. plot(risk,'b-')
23. xlabel('123 家企业')
24. ylabel('信贷风险量化指标')
25. legend('信贷风险')
26. title('123 家企业的信贷风险')
27.

```

```

28. function W = EntropyMethod(Z)
29. [n,m] = size(Z);
30. D = zeros(1,m);
31. for i = 1:m
32.     x = Z(:,i);
33.     p = x/sum(x);
34.     e = -sum(p.*my_log(p))/my_log(n);
35.     D(i) = 1-e;
36. end
37. W = D./sum(D);
38. end
39.
40. function lnp = my_log(p)
41. n = size(p,1);
42. lnp = zeros(n,1);
43. for i = 1:n
44.     if p(i) == 0
45.         lnp(i) = 0;
46.     else
47.         lnp(i) = log(p(i));
48.     end
49. end
50. end

```

问题 1.lg4 优化模型求解代码

```

!约束条件;
@for(cusA(i):@sum(rate(j):rA(i,j))=1);
@for(cusB(i):@sum(rate(j):rB(i,j))=1);
@for(cusC(i):@sum(rate(j):rC(i,j))=1);
@sum(cusA(i):xa(i))+@sum(cusB(i):xb(i))+@sum(cusC(i):xc(i))=6000;
@for(cusA(i):xa(i)<=100);
@for(cusA(i):xa(i)>=10);
@for(cusB(i):xb(i)<=100);
@for(cusB(i):xb(i)>=10);
@for(cusC(i):xc(i)<=100);
@for(cusC(i):xc(i)>=10);

@for(cusA(i):@for(rate(j)|j#GE#7:rA(i,j)=0));
@for(cusB(i):@for(rate(j)|j#LT#7 #or# j#GE#15:rB(i,j)=0));

```

```

@for(cusC(i):@for(rate(j)|j#LT#15 #or# j#GE#21:rC(i,j)=0));

!@for(cusA(i):@for(rate(j)|j#GE#7:rA(i,j))=0);
!@for(cusB(i):@sum(rate(j)|j#LT#7 #or# j#GE#15:rB(i,j))=0);
!@for(cusC(i):@sum(rate(j)|j#LT#15 #and# j#GE#21:rC(i,j))=0);

!@for(cusA(i):@sum(rate(j):rA(i,j)*LA(j))=0.0945741);
!@for(cusB(i):@sum(rate(j):rB(i,j)*LB(j))=0.0667996);
!@for(cusC(i):@sum(rate(j):rC(i,j)*LC(j))=0.0687253);

@for(cusA(i):xa(i)<=100);
@for(cusA(i):xa(i)>=90);
@for(cusB(i):xb(i)>70+10*(0.01-pb(i))/@abs(pb(i)-0.01));
@for(cusB(i):xb(i)<85+10*(0.01-pb(i))/@abs(pb(i)-0.01));
@for(cusC(i):xc(i)>15+5*(0.05-pc(i))/@abs(pc(i)-0.05));
@for(cusC(i):xc(i)<25+5*(0.05-pc(i))/@abs(pc(i)-0.05));

@for(linkA(i,j):@bin(rA(i,j)));
@for(linkB(i,j):@bin(rB(i,j)));
@for(linkC(i,j):@bin(rC(i,j)));
data:
pa = @ole("C:\Users\13593\Desktop\result (1).xlsx",'pa');
pb = @ole("C:\Users\13593\Desktop\result (1).xlsx",'pb');
pc = @ole("C:\Users\13593\Desktop\result (1).xlsx",'pc');
LA = @ole("C:\Users\13593\Desktop\result (1).xlsx",'LA');
LB = @ole("C:\Users\13593\Desktop\result (1).xlsx",'LB');
LC = @ole("C:\Users\13593\Desktop\result (1).xlsx",'LC');
L = @ole("C:\Users\13593\Desktop\result (1).xlsx",'L');

!输出分配贷款金额;

@text('x_result.txt')=@writefor(customerA(i):xa(i),' ');
@text('x_result.txt')=@writefor(customerB(i):xb(i),' ');
@text('x_result.txt')=@writefor(customerC(i):xc(i),' ');

!输出利润率矩阵;
@text('rate_result.txt')=@writefor(cusA(i):@writefor(rate(j):rA(i,j),'
'),@newline(1));
@text('rate_result.txt')=@writefor(cusB(i):@writefor(rate(j):rB(i,j),'
'),@newline(1));
@text('rate_result.txt')=@writefor(cusC(i):@writefor(rate(j):rC(i,j),'

```

```
'),@newline(1));
```

```
enddata
```

```
END
```

Problem2_2.m 决策树对于是否违约分类代码

```
1. clc
2. clear all;
3.
4. %% 预测信誉等级
5. %第一列是否违约 第二列信誉评级
6. company_info = xlsread('附件1_new2.xlsx',1,'C2:D124');
7. data = xlsread('问题一指标.xlsx',1,'B2:J124');
8. data_predict = xlsread('问题二指标.xlsx',1,'B2:H303');
9. n1 = size(data,1);
10. n2 = size(data_predict,1);
11. %分类模型 logistic 回归 是否违约为自变量 其他为因变量
12. [R] = corrcoef([data(:,2:end)]);
13. Y2 = data(:,1); %信誉等级 1 2 3 4
14. Y = data(:,2); %是否违约
15. X = data(:,3:end); %第一列为全部发票净利率 第二列不同代号个数 第三列单个代号重复次数均值 第七列为无效发票和负数发票的值
16. %第四列 2018-2019 总营业额 第五列 2018-2019 营业额增长率 第六列 2018-2019 净利润增长率 第七列是无效发票和负数发票
17. % [B,FitInfo] = lasso(X,Y,'CV',10,'PredictorNames',{'x1','x2','x3','x4','x5','x6','x7'});
18. % idxLambda = FitInfo.Index1SE;
19. % minMseModel = FitInfo.PredictorNames(B(:,idxLambda)~=0); %x1 总里程 x4 预计承运时间
20. % lassoPlot(B,FitInfo,'PlotType','CV');
21. % legend('show')
22. %随机选取训练集和测试集
23. rnd1 = randperm(n1);
24. train = rnd1(1:100);
25. test = rnd1(101:end);
26. %训练数据
27. train_x = X(train,:);
28. train_y = Y(train,:);
29. %测试数据
30. test_x = X(test,:);
31. test_y = Y(test,:);
```

```

32.
33. %创建决策树分类器
34. tree = ClassificationTree.fit(train_x,train_y);
35.
36. %查看决策数视图
37. view(tree);
38. view(tree,'mode','graph');
39.
40. %运用测试集预测
41. test_predict = predict(tree,test_x);
42.
43. %结果分析
44. illegal_num = length(find(train_y==1)); %违约
45. legal_num = length(find(train_y==0)); %没有违约
46. illegal_rate = illegal_num/100;
47. legal_rate = legal_num/100;
48. all_illegal = length(find(Y==1));
49. all_legal = length(find(Y==0));
50. test_illegal = length(find(test_y==1));
51. test_legal = length(find(test_y==0));
52. predict_num1 = length(find(test_predict==1 & test_y==1)); %预测违约
53. predict_num2 = length(find(test_predict==0 & test_y==0)); %预测没有违约
54. disp([' 样本总数:',num2str(n1),' 违约数:',num2str(all_illegal),' 没有违约数:',num2str(all_legal)]);
55. disp([' 训练集数量:',num2str(100),' 违约数:',num2str(illegal_num),' 没有违约数:',num2str(legal_num)]);
56. disp([' 测试集数量',num2str(n1-100),' 违约数',num2str(test_illegal),' 没有违约数',num2str(test_legal)]);
57. disp([' 预测违约数 ',num2str(predict_num1),' 错判数:',num2str(test_illegal-predict_num1),' 正确率=',num2str(100*predict_num1/test_illegal),'%'])
58. disp([' 预测没有违约数 ',num2str(predict_num2),' 错判数 ',num2str(test_legal-predict_num2),' 正确率=',num2str(100*predict_num2/test_legal),'%'])
59. leafs = logspace(1,2,10);
60.
61. N = numel(leafs);
62.
63. error = zeros(N,1);
64. for i = 1:N
65.     desicion_tree = ClassificationTree.fit(train_x,train_y,'crossval','on','minleaf',leafs(i));
66.     error(i) = kfoldLoss(desicion_tree);

```

```

67. end
68. plot(leafs,error)
69. xlabel('叶子节点包含的最小样本数')
70. ylabel('交叉验证误差')
71. title('叶子节点含有的最小样本数对于决策数的影响')
72.
73.
74. %得到化后决策数
75. bestTree = ClassificationTree.fit(train_x,train_y,'minleaf',leafs(find(error==min(
    error),1)));
76. view(bestTree,'mode','graph')
77.
78. %重新计算
79. best_res = resubLoss(bestTree);
80. best_err = kfoldLoss(crossval(bestTree));
81.
82. pre_res = resubLoss(tree);
83. pre_err = kfoldLoss(crossval(bestTree));
84.
85. %剪枝
86. [~,~,~,bestlevel] = cvLoss(tree,'subtrees','all','treesize','min');
87. best_tree = prune(tree,'Level',bestlevel);
88. view(best_tree,'mode','graph')
89.
90. resu = resubLoss(best_tree);
91. loss = kfoldLoss(crossval(best_tree));
92.
93. predict_1 = predict(best_tree,X);
94. all_illegal = length(find(Y==1)); %原始数据违约个数
95. all_legal = length(find(Y==0)); %原始数据没有违约个数
96. prec_illegal = length(find(Y==1 & predict_1==1)); %正确预测违约个数
97. prec_legal = length(find(Y==0 & predict_1==0)); %正确预测没有违约个数
98. mis_num1 = all_illegal-prec_illegal; %违约错判数
99. mis_num2 = all_legal-prec_legal; %没有违约错判数
100. acc_rate1 = 100*prec_illegal/all_illegal;
101. acc_rate2 = 100*prec_legal/all_legal;
102. disp(['剪枝后的决策数对于违约的错判数为:',num2str(mis_num1),' 违约错判率
    为:',num2str(acc_rate1)])
103. disp(['剪枝后的决策数对于没有违约的错判数为',num2str(mis_num2),' 违约错判率为
    ',num2str(acc_rate2)])
104.

```



```

105. final_predict = predict(best_tree,data_predict);
106. sum(final_predict)
107. disp(['附件 2 的预测违约数为:',num2str(sum(final_predict))])

```

Problem2_3.m 决策树对于 A,B,C 信誉等级分类代码

```

1. clc
2. clear all;
3.
4. %将没有违约的数据进行三分类
5. data = xlsread('问题一指标.xlsx',1,'B2:J124');
6. data_predict = xlsread('问题二指标.xlsx',1,'B2:H303');
7. is = xlsread('问题二指标.xlsx',1,'K2:K303');
8. data_predict = [data_predict is];
9. data = data(data(:,2)==0,:); %只包含没有违约的训练数据
10. data_prec = data_predict(data_predict(:,end)==0,:); %只包含没有违约的预测数据
11. n1 = size(data,1);
12. n2 = size(data_prec,1);
13. data(:,2) = [];
14. data_prec(:,end) = [];
15. Y = data(:,1); %信誉等级 1 2 3 -> A B C
16. X = data(:,2:end);
17. %随机选取训练集和测试集
18. rnd1 = randperm(n1);
19. train = rnd1(1:76);
20. test = rnd1(77:end);
21. %训练数据
22. train_x = X(train,:);
23. train_y = Y(train,:);
24. test_x = X(test,:);
25. test_y = Y(test,:);
26.
27. %创建决策树分类器
28. tree = ClassificationTree.fit(train_x,train_y);
29.
30. %查看决策数视图
31. view(tree);
32. view(tree,'mode','graph');
33.
34. %运用测试集预测
35. test_predict = predict(tree,test_x);
36. raw_1 = length(find(test_y==1)); %原始数据情况

```

```

37. raw_2 = length(find(test_y==2));
38. raw_3 = length(find(test_y==3));
39. prec_1 = length(find(test_predict==1)); %预测情况
40. prec_2 = length(find(test_predict==2));
41. prec_3 = length(find(test_predict==3));
42. acc_1 = length(find(test_y==1 & test_predict==1)); %正确数
43. acc_2 = length(find(test_y==2 & test_predict==2));
44. acc_3 = length(find(test_y==3 & test_predict==3));
45. score1 = 100*acc_1/raw_1;
46. score2 = 100*acc_2/raw_2;
47. score3 = 100*acc_3/raw_3;
48. disp(['对信誉等级 A 的分类准确率为:',num2str(score1),'% 对信誉等级 B 的分类准确率为:',num2str(score2),'% 对信誉等级 C 的分类准确率为:',num2str(score3),'%'])
49.
50. leafs = logspace(1,2,10);
51.
52. N = numel(leafs);
53.
54. error = zeros(N,1);
55. for i = 1:N
56.     desicion_tree = ClassificationTree.fit(train_x,train_y,'crossval','on','minleaf',leafs(i));
57.     error(i) = kfoldLoss(desicion_tree);
58. end
59. plot(leafs,error)
60. xlabel('叶子节点包含的最小样本数')
61. ylabel('交叉验证误差')
62. title('叶子节点含有的最小样本数对于决策数的影响')
63.
64. %得到化后决策数
65. bestTree = ClassificationTree.fit(train_x,train_y,'minleaf',leafs(find(error==min(error),1)));
66. view(bestTree,'mode','graph')
67.
68. %重新计算
69. best_res = resubLoss(bestTree);
70. best_err = kfoldLoss(crossval(bestTree));
71.
72. pre_res = resubLoss(tree);
73. pre_err = kfoldLoss(crossval(bestTree));
74.

```

```

75. %剪枝
76. [~,~,~,bestlevel] = cvLoss(tree,'subtrees','all','treesize','min');
77. best_tree = prune(tree,'Level',bestlevel);
78. view(best_tree,'mode','graph')
79.
80. resu = resubLoss(best_tree);
81. loss = kfoldLoss(crossval(best_tree));
82.
83. final_prec = predict(best_tree,data_prec);
84. result = 4*ones(302,1);
85. index = find(data_predict(:,end)==0);
86. result(index) = final_prec;
87.
88. %多次运行确定分类结果
89. mult_result = xlsread('问题二指标.xlsx',1,'L2:U303');
90. n = size(mult_result,1);
91. class_result = zeros(n,1);
92. for i = 1:n
93.     num1 = sum(mult_result(i,')==1);
94.     num2 = sum(mult_result(i,')==2);
95.     num3 = sum(mult_result(i,')==3);
96.     num4 = sum(mult_result(i,')==4);
97.     max_num = max([num1 num2 num3 num4]);
98.     if max_num == num1
99.         class_result(i) = 1;
100.    elseif max_num == num2
101.        class_result(i) = 2;
102.    elseif max_num == num3
103.        class_result(i) = 3;
104.    elseif max_num == num4
105.        class_result(i) = 4;
106.    end
107. end

```

问题 2.lg4 问题而优化模型求解

Model:

sets:

```

customerA/1..31/:xa,pa;
customerB/1..209/:xb,pb;
customerC/1..21/:xc,pc;
cusA/1..31/;

```

```

cusB/1..209/;
cusC/1..21/;
rate/1..29/:LA, LB, LC, L;
linkA(cusA, rate):rA;
linkB(cusB, rate):rB;
linkC(cusC, rate):rC;
endsets
!min = (a1*31+a2*209+a3*21)/261;
a1 = @sum(cusA(i):@sum(rate(j):rA(i,j)*LA(j)))/31;
a2 = @sum(cusB(i):@sum(rate(j):rB(i,j)*LB(j)))/209;
a3 = @sum(cusC(i):@sum(rate(j):rC(i,j)*LC(j)))/21;
(a1*31+a2*209+a3*21)/261<=0.44;
max = z1+z2+z3;
z1 = @sum(cusA(i):@sum(rate(j):(1-pa(i))*xa(i)*rA(i,j)*L(j)));
z2 = @sum(cusB(i):@sum(rate(j):(1-pb(i))*xb(i)*rB(i,j)*L(j)));
z3 = @sum(cusC(i):@sum(rate(j):(1-pc(i))*xc(i)*rC(i,j)*L(j)));
!z1+z2+z3>=600;
!约束条件;
@for(cusA(i):@sum(rate(j):rA(i,j))=1);
@for(cusB(i):@sum(rate(j):rB(i,j))=1);
@for(cusC(i):@sum(rate(j):rC(i,j))=1);
@sum(cusA(i):xa(i))+@sum(cusB(i):xb(i))+@sum(cusC(i):xc(i))=10000;

@for(cusA(i):xa(i)<=100);
@for(cusA(i):xa(i)>=10);
@for(cusB(i):xb(i)<=100);
@for(cusB(i):xb(i)>=10);
@for(cusC(i):xc(i)<=100);
@for(cusC(i):xc(i)>=10);

@for(cusA(i):xa(i)<=90);
@for(cusA(i):xa(i)>=80);
@for(cusB(i):xb(i)>25+5*(0.01-pb(i))/@abs(pb(i)-0.01));
@for(cusB(i):xb(i)<70+15*(0.01-pb(i))/@abs(pb(i)-0.01));
@for(cusC(i):xc(i)>15+5*(0.05-pc(i))/@abs(pc(i)-0.05));
@for(cusC(i):xc(i)<20+5*(0.05-pc(i))/@abs(pc(i)-0.05));

@for(cusA(i):@for(rate(j)|j#GE#7:rA(i,j)=0));
@for(cusB(i):@for(rate(j)|j#LT#7 #or# j#GE#15:rB(i,j)=0));
@for(cusC(i):@for(rate(j)|j#LT#15 #or# j#GE#21:rC(i,j)=0));

```

```

@for(linkA(i,j):@bin(rA(i,j)));
@for(linkB(i,j):@bin(rB(i,j)));
@for(linkC(i,j):@bin(rC(i,j)));

data:
pa = @ole("C:\Users\13593\Desktop\result2.xlsx",'pa');
pb = @ole("C:\Users\13593\Desktop\result2.xlsx",'pb');
pc = @ole("C:\Users\13593\Desktop\result2.xlsx",'pc');
LA = @ole("C:\Users\13593\Desktop\result2.xlsx",'LA');
LB = @ole("C:\Users\13593\Desktop\result2.xlsx",'LB');
LC = @ole("C:\Users\13593\Desktop\result2.xlsx",'LC');
L = @ole("C:\Users\13593\Desktop\result2.xlsx",'L');
!输出分配贷款金额;

@text('x_result.txt')=@writefor(customerA(i):xa(i),' ');
@text('x_result.txt')=@writefor(customerB(i):xb(i),' ');
@text('x_result.txt')=@writefor(customerC(i):xc(i),' ');

!输出利润率矩阵;
@text('rate_result.txt')=@writefor(cusA(i):@writefor(rate(j):rA(i,j),'
'),@newline(1));
@text('rate_result.txt')=@writefor(cusB(i):@writefor(rate(j):rB(i,j),'
'),@newline(1));
@text('rate_result.txt')=@writefor(cusC(i):@writefor(rate(j):rC(i,j),'
'),@newline(1));
enddata
END

```

Problem3_1.m 问题三量化信贷风险 and 数据处理代码

```

1. clc
2. clear all;
3.
4. data1 = xlsread('问题三指标.xlsx',1,'I2:I303'); %信誉等级
5. data2 = xlsread('问题三指标.xlsx',1,'B2:H303'); %常规数据
6. data3 = xlsread('问题三指标.xlsx',1,'K2:K303'); %疫情影响数据
7. data = [data1 data2 data3];
8. %第一列信誉等级 第二列总发票净利润率 第三列不同代号个数 第四列代号重复次数均值
9. %第五列 2018-2019 总营业额 第六列 2018-2019 营业额增长率
10. %第七列 2018-2019 净利润率 第八列无效与负数发票数 第九列疫情对经济影响(越大越好)

```

```

11. n = size(data,1);
12. std_data = (data-repmat(min(data),n,1))./(repmat(max(data),n,1)-
    repmat(min(data),n,1));
13.
14. data(:,1) = max(data(:,1))-data(:,1);
15. std_data = (data-repmat(min(data),n,1))./(repmat(max(data),n,1)-
    repmat(min(data),n,1));
16. weight = EntropyMethod(std_data);
17. D_P = sum([(repmat(max(std_data),n,1)-std_data).^2].*repmat(weight,n,1),2).^0.5;
18. D_N = sum([(repmat(min(std_data),n,1)-std_data).^2].*repmat(weight,n,1),2).^0.5;
19. score = D_N./(D_P+D_N); %分数越高风险越小
20. risk = max(score)-score; %调整为正向 分数越高风险越大
21. plot(risk,'b-')
22. xlabel('302 家企业')
23. ylabel('信贷风险量化指标')
24. legend('信贷风险')
25. title('302 家企业的信贷风险')
26.
27. %计算优化模型参数
28. result = xlsread('result3_1.xlsx',2,'F2:G303');
29. ret = result(result(:,1)~=4,:);
30. retA = ret(ret(:,1)==1,:);
31. retB = ret(ret(:,1)==2,:);
32. retC = ret(ret(:,1)==3,:);

```

问题 3.1.lg4 问题三一优化模型求解

Model:

sets:

```

customerA/1..31/:xa,pa;
customerB/1..209/:xb,pb;
customerC/1..21/:xc,pc;
cusA/1..31/;
cusB/1..209/;
cusC/1..21/;
rate/1..29/:LA, LB, LC, L;
linkA(cusA, rate):rA;
linkB(cusB, rate):rB;
linkC(cusC, rate):rC;

```

endsets

```

min = (a1*31+a2*209+a3*21)/261;

```

```

a1 = @sum(cusA(i):@sum(rate(j):rA(i,j)*LA(j)))/31;
a2 = @sum(cusB(i):@sum(rate(j):rB(i,j)*LB(j)))/209;
a3 = @sum(cusC(i):@sum(rate(j):rC(i,j)*LC(j)))/21;
!(a1*31+a2*209+a3*21)/261<=0.4;
!max = z1+z2+z3;

z1 = @sum(cusA(i):@sum(rate(j):(1-pa(i))*xa(i)*rA(i,j)*L(j)));
z2 = @sum(cusB(i):@sum(rate(j):(1-pb(i))*xb(i)*rB(i,j)*L(j)));
z3 = @sum(cusC(i):@sum(rate(j):(1-pc(i))*xc(i)*rC(i,j)*L(j)));
z1+z2+z3>=600;

!约束条件;
@for(cusA(i):@sum(rate(j):rA(i,j))=1);
@for(cusB(i):@sum(rate(j):rB(i,j))=1);
@for(cusC(i):@sum(rate(j):rC(i,j))=1);
@sum(cusA(i):xa(i))+@sum(cusB(i):xb(i))+@sum(cusC(i):xc(i))=10000;
@for(cusA(i):xa(i)<=100);
@for(cusA(i):xa(i)>=10);
@for(cusB(i):xb(i)<=100);
@for(cusB(i):xb(i)>=10);
@for(cusC(i):xc(i)<=100);
@for(cusC(i):xc(i)>=10);

!不同信誉等级的利润率约束;
@for(cusA(i):@for(rate(j)|j#GE#7:rA(i,j)=0));
@for(cusB(i):@for(rate(j)|j#LT#7 #or# j#GE#15:rB(i,j)=0));
@for(cusC(i):@for(rate(j)|j#LT#15 #or# j#GE#21:rC(i,j)=0));

@for(cusA(i):xa(i)<=90);
@for(cusA(i):xa(i)>=80);
@for(cusB(i):xb(i)>25+10*(0.01-pb(i))/@abs(pb(i)-0.01));
@for(cusB(i):xb(i)<70+10*(0.01-pb(i))/@abs(pb(i)-0.01));
@for(cusC(i):xc(i)>15+5*(0.05-pc(i))/@abs(pc(i)-0.05));
@for(cusC(i):xc(i)<20+5*(0.05-pc(i))/@abs(pc(i)-0.05));

@for(linkA(i,j):@bin(rA(i,j)));
@for(linkB(i,j):@bin(rB(i,j)));
@for(linkC(i,j):@bin(rC(i,j)));
data:
pa = @ole("C:\Users\13593\Documents\Tencent
Files\2794459535\FileRecv\result3_1.xlsx",'pa');

```

```

pb = @ole('C:\Users\13593\Documents\Tencent
Files\2794459535\FileRecv\result3_1.xlsx','pb');
pc = @ole('C:\Users\13593\Documents\Tencent
Files\2794459535\FileRecv\result3_1.xlsx','pc');
LA = @ole('C:\Users\13593\Documents\Tencent
Files\2794459535\FileRecv\result3_1.xlsx','LA');
LB = @ole('C:\Users\13593\Documents\Tencent
Files\2794459535\FileRecv\result3_1.xlsx','LB');
LC = @ole('C:\Users\13593\Documents\Tencent
Files\2794459535\FileRecv\result3_1.xlsx','LC');
L = @ole('C:\Users\13593\Documents\Tencent
Files\2794459535\FileRecv\result3_1.xlsx','L');
!输出分配贷款金额;

@text('x_result.txt')=@writefor(customerA(i):xa(i),' ');
@text('x_result.txt')=@writefor(customerB(i):xb(i),' ');
@text('x_result.txt')=@writefor(customerC(i):xc(i),' ');

!输出利润率矩阵;
@text('rate_result.txt')=@writefor(cusA(i):@writefor(rate(j):rA(i,j),'
'),@newline(1));
@text('rate_result.txt')=@writefor(cusB(i):@writefor(rate(j):rB(i,j),'
'),@newline(1));
@text('rate_result.txt')=@writefor(cusC(i):@writefor(rate(j):rC(i,j),'
'),@newline(1));

enddata
END

```

Output_result.m 数据导出代码

```

1. clc
2. clear all;
3.
4. %计算 lingo 结果
5. load x_result.txt;
6. load rate_result.txt;
7.
8. x_result = x_result';
9. profit_rate = [0.04 0.0425 0.0465 0.0505 0.0545 0.0585 0.0625 0.0665 0.0705 0.0745
    0.0785 0.0825 0.0865 0.0905 0.0945 0.0985 0.1025 0.1065 0.1105 0.1145 0.1185 0.12
    25 0.1265 0.1305 0.1345 0.1385 0.4125 0.1465 0.15]';
10.

```



```

11. %计算填入的位置
12. data = xlsread('result2.xlsx',2,'F2:F303');
13. pos_D = find(data==4); %x=0 r=0
14. pos_ABC = find(data~=4);
15.
16. rate_ABC = zeros(size(rate_result,1),1);
17. for i = 1:size(rate_result,1)
18.     obj = rate_result(i,:);
19.     rate_ABC(i) = profit_rate(find(obj==1));
20. end
21.
22. %填充结果
23. n = size(data,1);
24. final_x = zeros(n,1);
25. final_rate = zeros(n,1);
26. final_x(pos_D) = 0;
27. final_x(pos_ABC) = x_result;
28. final_rate(pos_D) = 0;
29. final_rate(pos_ABC) = rate_ABC;

```

Range_analysis.m 灵敏度分析

```

1. clc
2. clear all;
3.
4. %灵敏度分析
5. %控制流失率 优化收益
6. range1 = xlsread('灵敏度分析数据.xlsx',1,'D1:K2');
7. figure(1)
8. plot(range1(1,:),range1(2,:), 'b-')
9. xlabel('控制流失率')
10. ylabel('最大化收益值')
11. title('控制流失率优化收益的灵敏度分析')
12.
13. %控制收益 优化流失率
14. range2 = xlsread('灵敏度分析数据.xlsx',1,'D10:K11');
15. figure(2)
16. plot(range2(1,:),range2(2,:), 'b-')
17. xlabel('控制收益')
18. ylabel('最小化流失率')
19. title('控制收益优化流失率的灵敏度分析')

```

