

赛区评阅编号：
(由赛区填写)

全国评阅编号：
(全国组委会填写)

2020 高教社杯全国大学生数学建模竞赛

编 号 专 用 页

赛区评阅记录 (可供赛区评阅时使用):

评 阅 人						
备 注						

送全国评阅统一编号：
(赛区组委会填写)

(请勿改动此页内容和格式。此编号专用页仅供赛区和全国评阅使用，参赛队打印后装订到纸质论文的第二页上。注意电子版论文中不得出现此页。)

基于 TOPSIS 与 KMV 模型的信贷决策优化模型

摘要

随着中小企业在社会中的贡献度不断上升，与其贷款难相关的信用问题得到越来越多的关注。由于中小企业的规模较小，资产与现金流不稳定，因此银行有必要对企业信誉进行评估，并据此制定信贷策略，以达到银行贷款风险降低，企业有款可贷的目的。

针对问题一：本文通过对数据的特征挖掘，发现了影响中小企业信贷风险的主要因素：企业毛利率、发票作废比例、总税额、信誉评级等，并为其设计了评分体系。由于数据量较大且信贷风险评分较为客观，因此本文通过熵权法对权重进行评估。又因为需要对所有企业进行整体信誉评估，且 TOPSIS 模型对数据整体性把握较好，在此被用于对企业信贷风险评分与排序。进一步地，通过 K-Means 算法对企业进行二分类，排除不可贷款企业。同时，由于对利率与银行客户流失率进行函数拟合并建立信贷配给模型，以银行收益最大化为目标构造函数并进行优化求解。通过模拟分析与理论求解，三种信誉情况下银行利率均存在最优解，分别为 **5.8%**、**6.1%** 和 **6.2%**，利率过大或过小均会使银行收益降低。此结果说明信誉更好的企业，银行可给予的利率优惠可以更大。最后，由于定额资金分配时企业数量较多，本文以风险最低建立目标函数，资金定额与企业需求作为约束因素进行优化，采用遗传算法对各企业贷款资金进行求解。

针对问题二：除无信誉评级标签之外，本文其余数据形式与问题一中相同，因此首先使用与第一问相似方法进行数据处理分析并对贷款金额精确求解。针对数据没有信誉标签的情况，我们使用决策树模型对其进行标签化分类。在此过程中，我们使用问题一中数据作为训练集并进行交叉验证，精确度为 **0.75**，AUC 值为 **0.62**，精确度较好。在此基础上对问题二数据集作为测试集进行预测。

针对问题三：由于突发因素对不同行业的中小企业影响为即时影响且有所区别，本文使用 KMV 模型以将即时影响因素纳入考量，并将其转化为对相关行业股市指数变化的影响。进一步，可据此计算预期违约概率变化。结果显示（以新冠疫情为例），服务业、个体营业者、制造业受冲击最大，违约概率分别从 **0.554**、**0.729**、**0.623** 增加到了 **0.603**、**0.986**、**0.763**；食品、IT 行业受冲击次之，生物医药产业发展最好。最后讲违约概率纳入已有 TOPSIS 评分模型中进行修正，并据此给出了问题二中所有企业的最新信贷策略。

关键字： 信贷配给 KMV 模型 TOPSIS 模型 决策树 遗传算法

一、问题重述

银行作为盈利性组织，在制定信贷政策过程中会权衡成本与收益。相比国有大中型企业有大量的固定资产用于抵押，中小微企业大多是自负盈亏的民营企业且规模较小缺少抵押资产。银行依据信贷政策与企业财务数据对企业的实力，供需关系，信誉与信贷风险做出提供贷款与利率优惠的判断。中小微企业的实力、信誉作为其信贷风险评估的依据，然后根据信贷风险确定是否进行信贷业务及信贷金额、利率和期限。建立数学模型解决以下问题：

1. 根据信贷记录对 123 家企业的信贷风险进行量化并根据信贷风险给出在信贷总额固定的情况下，不同信贷风险企业的信贷策略。
2. 假设银行信贷总额为 1 亿元，对 302 家无信贷记录的企业的信贷风险进行量化分析，并根据信贷风险确定信贷策略。
3. 企业在实际生产经营中会受到一些突发因素的影响，而且突发因素对不同行业、类别的企业产生的影响具有差异性。因此在加入突发因素的影响下，考虑 302 家企业的信贷风险，并给出在信贷总额为 1 亿元的信贷调整策略。

二、模型假设

1. 假设附件一与附件二中的企业进项发票与销项发票可充分表示企业财务状况与实力。
2. 假设年度信贷总额分配仅在附件已知企业内进行。
3. 假设银行为实现利润最大化，在考虑客户留存率的前提下，适当提高贷款年利率。

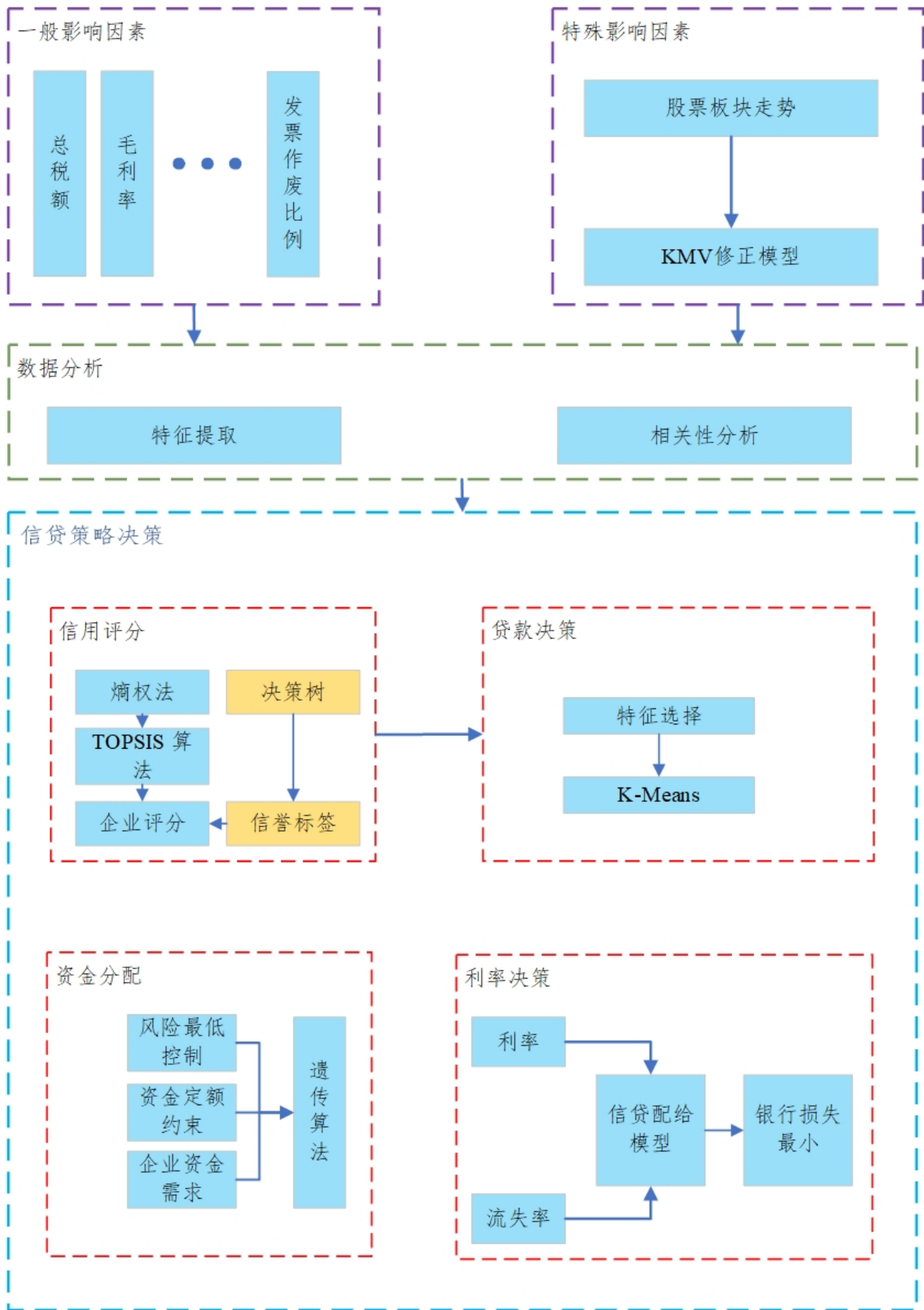


图 1 模型整体框架图

三、问题一

3.1 问题一的分析

在问题一中银行首先根据中小微企业财务状况的实力、信誉评级对信贷风险做出评估，并根据信贷风险做出贷款决策及贷款金额等信贷策略。针对问题一，我们首先利用熵权法确定权重，然后利用 TOPSIS 根据熵权法做出的影响因素的权重得出 123 家企业的信用评分。在得到 123 家企业的信用评分后，本模型利用无监督学习方法 K-Means 对是否放贷做出判断，根据风险最低控制、资金定额约束、企业资金需求使用遗传算法分配资金，最后根据信贷配给模型对利率做出判断。

3.2 数据预处理

由于题目提供的数据较为分散，因此本模型首先介绍如何进行数据清洗。由于企业的进项发票与销项发票同时反映着企业的经营财务状况，因此根据企业的进项发票与销项发票，利用企业代号进行聚合可得出企业的销售收入和采购支出。进一步，由销售收入与采购支出可得到企业毛收入与毛利率。同时根据发票中作废的数量可得到企业发票作废比例，这些都是反映企业信贷风险的重要指标。通过对这些指标进行相关性分析，如图2所示。最终选取企业毛收入与毛利率，发票作废比例，企业信用评级作为下一步基于熵权法的 TOPSIS 评价模型的指标。各个评价指标为：

- 毛收入 = 销售收入 - 采购支出
- 毛利率 = 毛收入 / 销售收入
- 发票作废比例 = 作废发票计数 / 总发票计数
- 企业信用评级为附件一中企业信用等级。

从图中可以看出：企业毛收入与毛利率，发票作废比例，企业信用评级之间相关性较低，所以我们选取这几个指标。

3.3 基于熵权法的修正 TOPSIS 模型对信贷风险的量化评分

3.3.1 指标分类

根据上文中的数据预处理提取的特征指标对其进行指标分类利于下文的指标正向化。在此基础上在这里为了评分方便对 A,B,C,D 的信誉评级做了一个数据映射处理关系如示。A、B、C、D 对应映射评分分别为 1、2、3、4。毛收入、毛利率、发票作废比例、信誉评级分别为极大型、极大型、极小型、极小型。

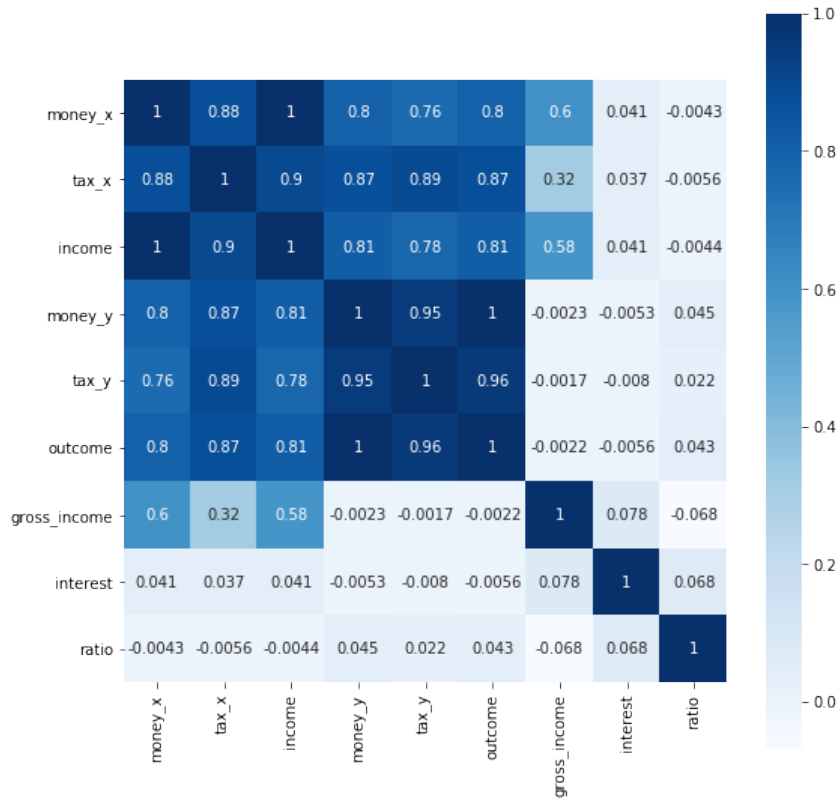


图 2 指标相关性分析

3.3.2 权重计算

由于本文中数据较多，且判断方法不宜主观（AHP 等），因此熵权法在这里被使用，可以有效避免各评价指标的认为因素干扰，更可以反应各个特征的原始信息。本次熵权法步骤如下：

1. 对参与评价的企业集合 M 为 $(1, 2, \dots, m)$ ，特征因素集 D 为 $(1, 2, \dots, n)$ ，那么不妨设 X_{ij} 为第 i 个企业的第 j 个特征的数值。

$$X = \begin{bmatrix} & D_1 & D_2 & \cdots & D_n \\ M_1 & x_{11} & x_{12} & \cdots & x_{1n} \\ M_2 & x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ M_m & x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \quad (1)$$

2. 为方便求解，将所有指标转化为正向指标，并对特征因素数据进行归一化，转为同质问题。

$$x_{ij} = \frac{x_{ij} - \min\{x_{1j}, \dots, x_{nj}\}}{\max\{x_{1j}, \dots, x_{nj}\} - \min\{x_{1j}, \dots, x_{nj}\}} \quad (2)$$

3. 计算特征 j 下第 i 个企业的比重：

$$p_{ij} = \frac{x_{ij}}{\sum_{i=1}^n x_{ij}}, \quad i = 1, \dots, n, j = 1, \dots, m \quad (3)$$

4. 计算第 j 项特征因素的熵

$$e_j = -k \sum_{i=1}^n p_{ij} \ln(p_{ij}), \quad j = 1, \dots, m \quad (4)$$

5. 计算信息熵冗余度与各特征因素的权重:

$$d_j = 1 - e_j, \quad j = 1, \dots, m \quad (5)$$

$$w_j = \frac{d_j}{\sum_{j=1}^m d_j}, \quad j = 1, \dots, m \quad (6)$$

3.3.3 计算信用评分

假设有 n 个要评价的对象， m 个评价指标构成正向化矩阵，如下:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \quad (7)$$

对矩阵每一个元素进行标准化:

$$z_{ij} = x_{ij} / \sqrt{\sum_{i=1}^n x_{ij}^2} \quad (8)$$

3.3.4 计算得分归一化

定义第 $i(i = 1, 2, \dots, n)$ 个评价对象与最大值的距离:

$$D_i^+ = \sqrt{\sum_{j=1}^m (Z_j^+ - z_{ij})^2} \quad (9)$$

定义第 $i(i = 1, 2, \dots, n)$ 个评价对象与最小值的距离:

$$D_i^- = \sqrt{\sum_{j=1}^m (Z_j^- - z_{ij})^2} \quad (10)$$

那么，计算得出第 $i(i = 1, 2, \dots, n)$ 个评价对象未归一化的得分:

$$S_i = \frac{D_i^-}{D_i^+ + D_i^-} \quad (11)$$

显然 $0 \leq S_i \leq 1$, 且 S_i 越大 D_i^+ 越小, 即越接近最大值。

3.3.5 信贷风险评分

通过基于熵权法修改的 TOPSIS 法，我们对原始数据进行了同趋势和归一化处理，消除了不同指标量纲的影响。排序的结果充分利用原始数据信息，能定量反映不同评价单元的优劣程度，较为直观、可靠。所得到的加权平均值。相对接近度值在 0 与 1 之间，该值愈接近 1，反映所评价单元接近最优水平程度愈高，反之，该值愈接近 0，评价单元接近最优水平的程度愈低或者说愈接近最劣水平。因此我们将 TOPSIS 的结果作为反映信贷风险的量化结果，该值愈接近 1，反映所评价单元信贷风险越低，该值愈接近 0，反映所评价单元信贷风险越高。TOSIS 评分结果与分布图如图3所示。

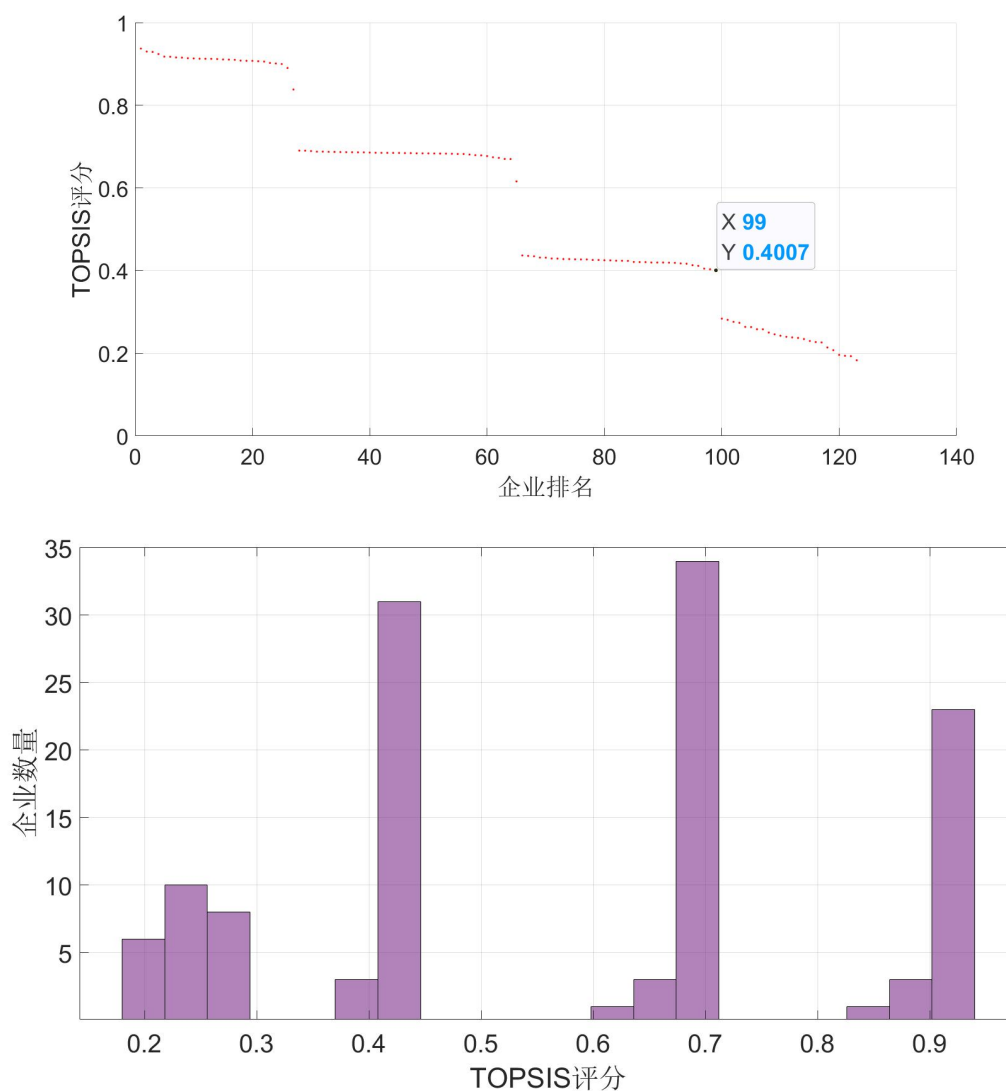


图 3 TOPSIS 信贷风险评分与分布

综上所述分析可知：

1. 在 TOPSIS 评分结果图可知，在评分 0.4 处有个明显的跳变，将其对应其信誉等级后发现其正好与信誉评级 C 与 D 的分界线处相对应。因此在此可将其值大小作为可否

信贷的阈值。也与题目中原则上不给予 D 等级贷款相互对应，在此基础上也能一定说明此阈值评分标准的精准性。

2. 在 TOPSIS 评分结果中，可发现少数的信誉等级为 C 的企业的评分是大于为 A 的企业，说明评价信誉等级并不能完全评判其评分大小，也为本文后面一些评判等级不是 D 的也不能贷款埋下铺垫。
3. 在评分结果图中也能看出高分是数量较少的，而低分也是数量较少的，此种分配能几乎满足类似正太分布的特点，也反映出其评分的合理性。
4. 此结果图中的跳变点的存在性也反映了此评分作为是否贷款的合理性。

3.4 基于信贷配给模型的利率确定

3.4.1 信贷配给模型因素以及目标函数确定

在附件三中，可以看出在不同信誉等级下，利率与客户的流失率以及有明显的关系如图4所示。本文基于此种关系，为寻找在此关系中的函数关系，将信誉评价作为其分段函数的条件、利率作为自变量、流失率考虑为因变量对其进行曲线拟合。在附件三中，

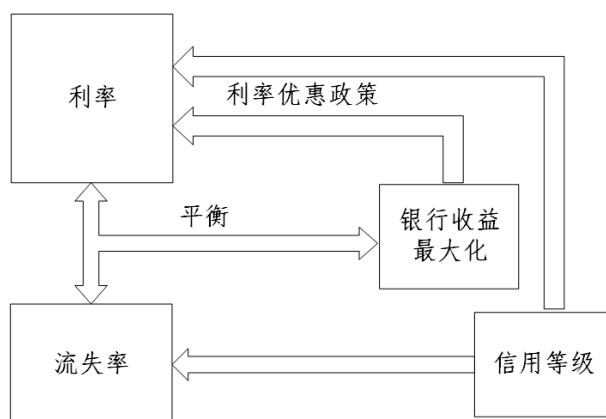


图 4 利率确定条件图

可以看出在不同信誉等级下，利率与客户的流失率以及有明显的关系如图 3 所示。本文基于此种关系，为寻找在此关系中的函数关系，将信誉评价作为其分段函数的条件、利率作为自变量、流失率考虑为因变量对其进行曲线拟合。随着利率的增加，客户流失率在随之增加。因此曲线拟合函数是完全合理的。公式为：

$$f(x) = a1 * \sin(b1 * x + c1) \quad (12)$$

此公式 x 为利率，f(x) 为流失率。在不同信誉等级下的公式中的值为：

$$\begin{cases} a1 = 0.8852; b1 = 14.27; c1 = -0.4492; & (\text{在信誉评价为A下}) \\ a1 = 0.851; ; b1 = 13.92; c1 = -0.4498; & (\text{在信誉评价为B下}) \\ a1 = 0.8539; b1 = 13.67; c1 = -0.4538; & (\text{在信誉评价为C下}) \end{cases} \quad (13)$$

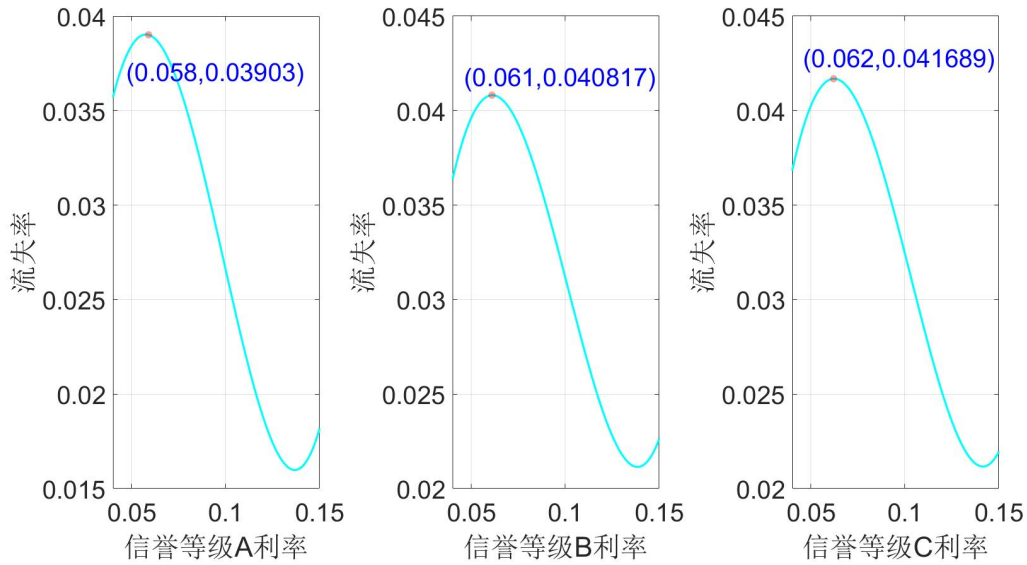


图 5 信誉等级为 A,B,C 时银行获利目标函数图

在此基础上构建关于银行利益最大化的信贷配给模型目标函数：

$$y(\text{利率}) = \text{利率} * (1 - \text{流失率}(\text{利率})) \quad (14)$$

3.4.2 最优值利率及结果分析

本文在不同信誉条件下，找出此函数最大值所在点处如图5所示，即为银行获利最大点。由上图可知在规定银行获利最大的条件下，信誉评级 A,B,C 下的利率分别为 5.8%，6.1%，6.2%。综上结果分析可知：

1. 利率减少，客户流失率增加。利率增加，客户流失率减小。若要银行取得最大值，需要在此基础上利率和流失率中寻找平衡。
2. 在规定利率区间里，在起初利率为 [4%,5.8%] 区间中，利率增加对于银行获利最多的影响大于流失率增加对其的影响。而在之后的利率为 [6%,14%]，流失率的增加的影响大于利率的增加的对其的影响。
3. 在此函数的最大值处的横坐标利率即为银行获利最多，即此时的平衡性最优。

3.5 K-Means 贷款决策模型

由于贷款与否的数据具有相同的特征，因此在本模型中使用无监督学习方法 K-Means 将用户数据进行聚类，分类为允许贷款与拒绝贷款。

3.5.1 模型的建立

1. 将数据集中每一个样本点被逐个指派到与其最近的中心的类中，求得一个划分，使得目标函数极小化：

$$\min_C \sum_{l=1}^k \sum_{C(i)=l} \|x_i - m_l\|^2 \quad (15)$$

2. 然后对于给定的划分 C，求得各个类的中心 $M = m_1, m_2$ ，使得目标函数极小化：

$$\min_{m_1, \dots, m_k} \sum_{l=1}^k \sum_{C(i)=l} \|x_i - m_l\|^2 \quad (16)$$

3. 最后更新每个类的样本均值，作为新的聚类中心：

$$m_l = \frac{1}{n_l} \sum_{C(i)=l} x_i, \quad l = 1, \dots, k \quad (17)$$

3.6 基于遗传算法的贷款额度判定

有学者从生物群落中的遗传现象中的到启发，将这种遗传机制应用到了算法中来进行计算。模拟生物群落中的遗传机理，在遗传算法中，通过二进制编码组成初始群落后，通过遗传操作使群落中的个体交配产生新的个体，再对新产生的个体按照他们对环境的适应度实现优胜劣汰，直到筛选出最优的一个个体，该个体即为待优化目标的最优解。

在此算法运算中，目标函数为风险函数，在风险函数最低的情况下，以银行放贷总额一定以及贷款额度的限制作为约束判断其各个企业贷款额度。风险函数构造如下：

$$W = \sum_{n=1}^n a_n x_n \quad (18)$$

由 TOPSIS 评分构成的风险因子的函数：

$$a_n = e^{-t_n} \quad (19)$$

t_n 为 TOPSIS 各项评分， a_n 为风险因子。

限制条件为：

$$\begin{cases} 10 \leq x_n \leq 100 \\ \sum_{n=1}^n x_n = \text{银行给定金额} \end{cases} \quad (20)$$

3.7 贷款额度

在此问题中运用 GA 算法工具箱设置银行放贷总额一定进行求解，而的出的结果即为本文所给出的贷款额度建议。此项过程运用了放贷定额的求解以及结果将在问题二以及问题三中体现。

四、问题二

4.1 问题二的分析

由于附件二未给出信誉评级，因此在本模型中，首先运用决策树预测附件二中 302 家企业的信誉评级。然后运用与问题一中相似的基于熵权法的 TOPSIS 模型得出 302 家企业的信贷风险评分，信贷风险评分排名如图6所示。并利用信贷风险评分等指标首先判断贷款与否，进一步利用问题一中提出的贷款年利率与客户流失率的关系取得最优值利率，最后通过 GA 工具箱求得在银行信贷总额固定为一亿条件下，对企业的信贷策略。

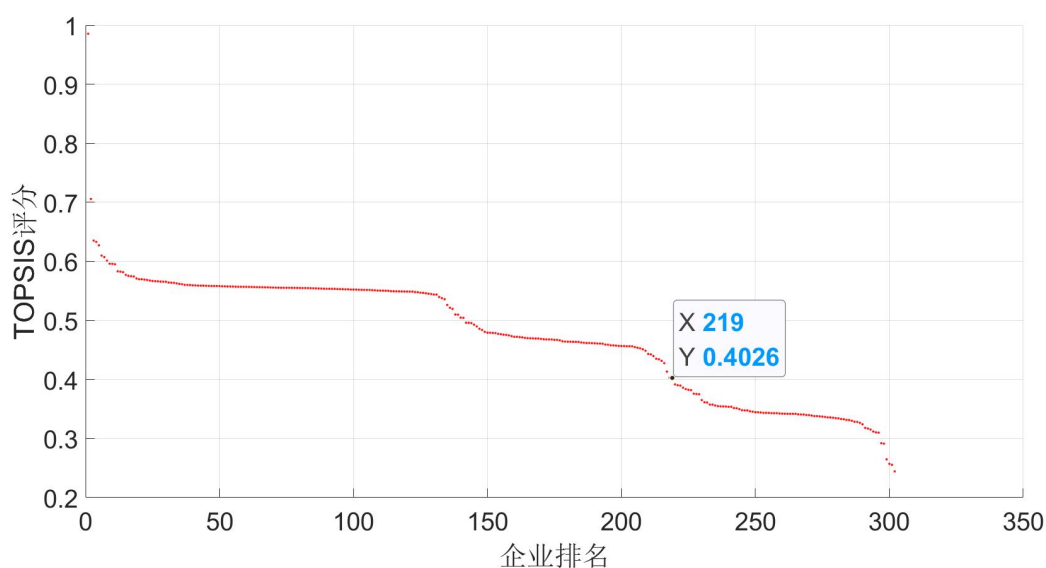


图 6 信贷风险评分排名

4.2 决策树算法确定信誉标签

由于在问题二中，302 家企业信贷数据没有给出，因此本模型首先采用决策树算法，利用附件一中给出的 123 家企业数据作为训练集，预测附件二中企业信贷数据。

4.2.1 模型建立

1. 对于数据集 D ，不同的特征存在条件信息熵 $\text{Info } A(|D|)$:

$$\begin{aligned} \text{Info}_4(D) &= - \sum_{k=1}^K \frac{|D_k|}{D} \times \text{Info}(D_j) \\ &= - \sum_{k=1}^K \frac{|D_k|}{D} \sum_{i=1}^I \frac{|D_k|}{D} \log_2 \frac{|C_k|}{D} \end{aligned} \quad (21)$$

2. 计算特征的信息增益:

$$\text{Gain}(A) = \text{Info}(D) - \text{Info}_A(DD) \quad (22)$$

3. 对于样本集合 D , 该样本的基尼系数定义为:

$$\text{Gini}(D) = 1 - \sum_{k=1}^K \left(\frac{|C_k|}{|D|} \right)^2 \quad (23)$$

4. 根据特征 A 划分的基尼系数为:

$$\text{Gini}(D, A) = \frac{|D_1|}{|D|} \text{Gini}(D_1) + \frac{|D_2|}{|D|} \text{Gini}(D_2) \quad (24)$$

4.2.2 模型求解

首先将附件一中的数据按照训练集: 测试集 = 8:2 进行划分, 测试精度为 0.72396, AUC 曲线如图7, 利用 Graphviz 可视化结果如图8所示。然后将训练模型应用于附件二, 得到 302 家企业的信贷等级。如图8所示, 决策树模型根据所选指标, 例如毛收入、毛

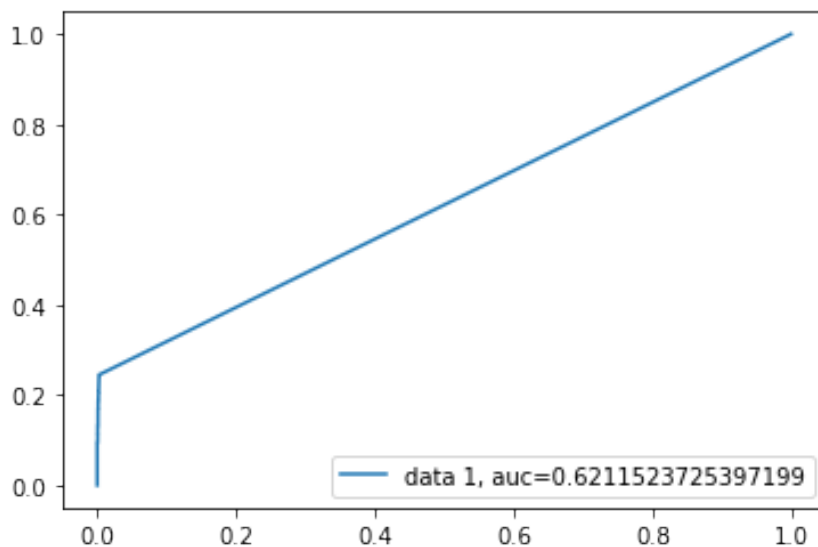


图 7 决策树模型在附件一中的预测结果 AUC 曲线

利润、发票作废比例等影响因素, 根据基于基尼系数的信息增益划分节点, 最终做出决策。

4.3 年度信贷总额固定时的信贷策略

4.3.1 模型求解

在确定是否进行贷款与根据信誉评级确定贷款年利率后, 使用 Matlab 自带的 GA 工具箱, 在对初始种群赋初值时, 应先对各参数的取值范围做一个粗略的估计, 在给各

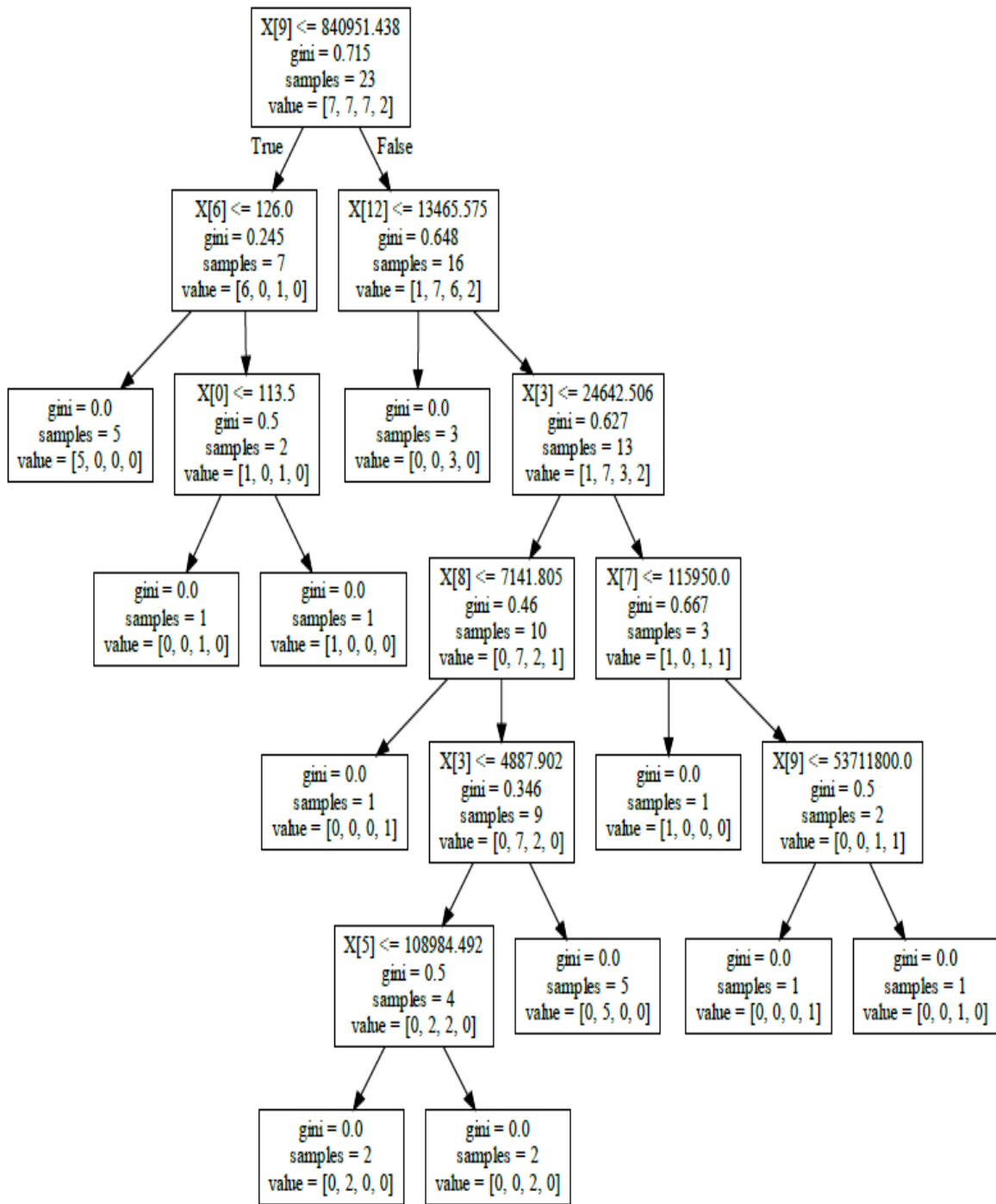


图 8 决策树模型判定信贷等级

参数赋初值时要尽量覆盖到这个取值范围，这样有利于提高算法的运行效率。对于比较复杂的目标函数，常常存在多个极值点，如果在给初始种群赋初值时未能覆盖各参数的取值范围或是初始种群离全局最优解较远，则算法很容易就陷入到局部最优解当中。设置种群个数为 50，迭代次数为 600 等参数，求解贷款额度直方图如图9所示。

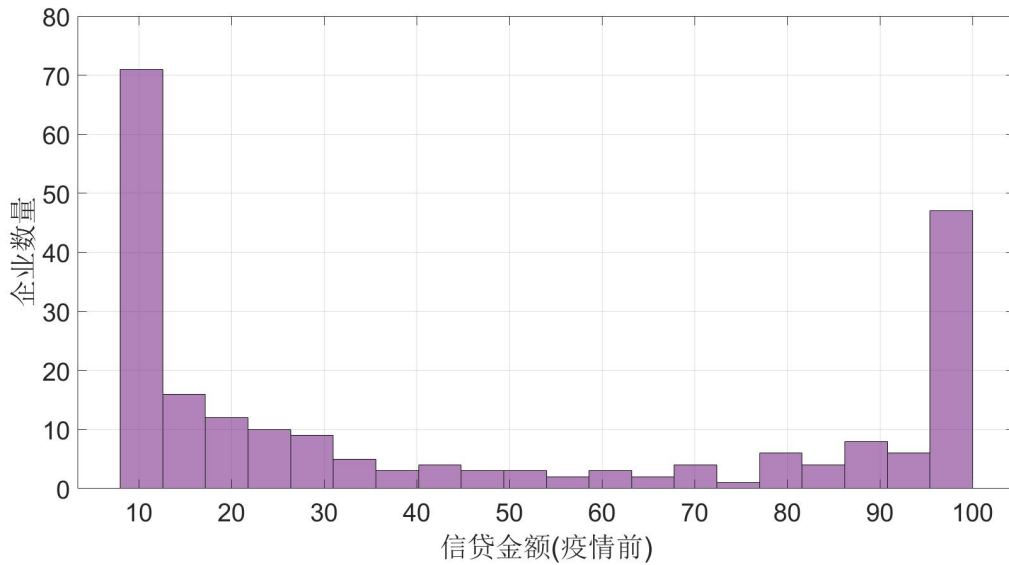


图 9 贷款金额分布直方图

4.3.2 结果分析

- 根据 K-Means 无监督学习结果，企业信贷风险评分排名 219 之后的企业不予贷款。如图6在信贷风险评分排名所示，与问题一中有违约记录的截断点对比可发现：信贷风险评分低于 0.4 时，银行不予贷款。
- 在贷款金额分布直方图中（图9），贷款金额与企业数量的关系成 U 字形。经分析可得，其为指数衰减曲线的拖尾现象。在微观市场竞争理论中，由于市场的完全竞争性质，企业数会随实力增长迅速衰减，却在较远处出现小幅集中的垄断竞争集团。而在进行数据清洗的过程中，我们也发现企业财务数据体现的现金流能力较强与较弱的企业占大多数，与例论和现实生活相符合。

五、问题三

5.1 问题三的分析

由于突发因素往往难以预测且对不同行业影响较大，因此有必要发生突发事件时对各行业企业进行及时评估，将信用变化情况纳入 TOPSIS 评分机制，进行信贷策略的调整。在突发性问题信用风险评估中，目前主流的方法为 KMV 模型与 Creditmetrics 模型。KMV 模型将企业贷款信用风险与股票同行业指数相联系，建立模型违约点并评估违约概率。CreditMetrics 模型则利用转换矩阵的输入，对风险变化进行评估。本文采用 KMV 模型，分析如下：

- 随着股市发展，行业板块可以较好反应该行业当下真实情况，因此复合 KMV 模型使用条件。

- KMV 模型的数据从公开数据中较容易获取与进行操作评估。
- KMV 模型结果不依赖信用评级，对突发事件产生的影响有更为客观的反应。
- 转换矩阵一般由信用评级公司提供，不公开情况下难以获得。
- CreditMetrics 模型依赖历史违约率，且直接采取历史平均值，对历史数据要求较高，因此不采用 CreditMetrics 模型。

5.2 KMV

1. 市值及波动率的计算。

由期权定价公式可得：

$$V_E = h(V_A, \sigma_A, r_f, D, T) = V_A N(d_1) - D e^{-r_f T} N(d_2) \quad (25)$$

此公式中：

$$d_1 = \frac{\ln\left(\frac{V_A}{D}\right) + \left(r_f + \frac{\sigma_A^2}{2}\right) T}{\sigma_A \sqrt{T}} \quad (26)$$

$$d_2 = d_1 - \sigma_A \sqrt{T} = \frac{\ln\left(\frac{V_A}{D}\right) + \left(r_f - \frac{\sigma_A^2}{2}\right) T}{\sigma_A \sqrt{T}} \quad (27)$$

其中， V_E 为股权价值， V_A 为公司资产市值。 σ_A 为市值波动率， r_f 为无风险利率，可从央行公布数据获得。 D 为公司负债， T 为公司债务周期。变量计算方法分别如下：

- 股权价值 $V_E = \text{股本数} \times \text{估价} + \text{非流通股数} \times \text{每股净资产}$
- 股票收益率。由于股票周收益率为 $\mu_t = \ln\left(\frac{P_t}{P_{t-1}}\right)$ ， P_t 为股票每周收盘价。一般取我国股票市场交易日为 250 个/年，则年波动率 σ_E 为：

$$\sigma_E = \sigma_i \sqrt{\frac{250}{5}} = \sqrt{\frac{50 \sum_{t=1}^n (\mu_t - \bar{\mu})^2}{n-1}} \quad (28)$$

2. 违约点的计算

$$DPT = STD + C \times LTD \quad (29)$$

式中，STD 为短期负债，LTD 为长期负债。由于根据凌江怀等学着的文章，C 常取值为 1，即选取负债总额 D 作为违约点。

3. 违约距离与违约概率的计算

$$DD = \frac{V_A - DPT}{V_A \times \sigma_A} \quad (30)$$

同时依据概率分布计算预期违约概率 EDF 为：

$$EDF = N(-DD) = 1 - N(DD) \quad (31)$$

式中，N 为标准正态分布函数

5.3 突发因素对年度信贷总额固定的信贷策略影响

在问题二的条件下，考虑到新型冠状病毒对其评分会造成影响。因此 TOPSIS 评分会发生改变。前文得到基于 KMV 模型给出了各个行业在此期间的违约率, 如图11所示。为了保证 TOPSIS 评分的正确率，所以在此问在 TOPSIS 评分项中增加各个行业在新型冠状病毒期间违约率这一指标进行评分。为了便于对比分析，此处给出新型冠状病毒前后 TOPSIS 对比评分图如图10所示。详细结果见支撑材料。利率在前文只与信誉等级有关，

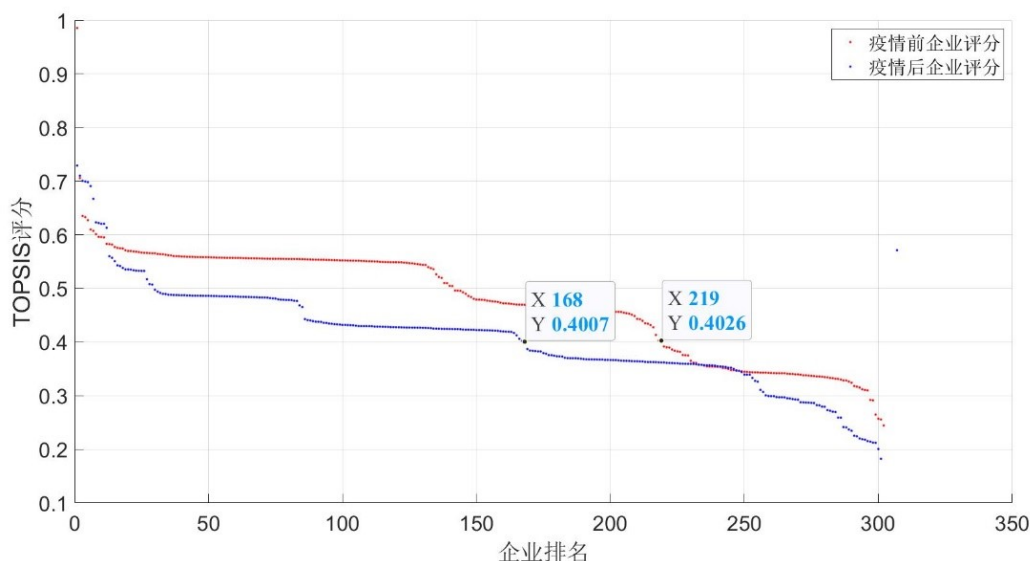


图 10 新型冠状病毒前后 TOPSIS 信贷风险评分对比图

而与 TOPSIS 评分无关，因此此问与二问的利率相同。

因为在此问中 TOPSIS 评分发生改变，所以在 GA 算法中的风险因子函数发生了改变，因此将会改变贷款额度。在此做出了新型冠状病毒前后的贷款额度分布如图12所示。详细结果见支撑材料。

5.4 模型结果分析

1. 从图10可看出在所有企业受到新型冠状病毒冲击后，其 TOPSIS 评分大体都是呈下降趋势，说明大部分行业都基本受到新型冠状病毒负面影响。
2. 从图10还可以得知在新型冠状病毒影响下，其评分为 0.55 到 0.70 的企业数是增多的，从其增多的企业可知其构成主要部分为生物医药与食品产业。侧面反映出新型冠状病毒促进了生物医药与 IT 产业的蓬勃发展。
3. 图10能看出能满足银行贷款要求的企业也从 219 家减少到了 168 家，而在银行放贷一定的情况下此充分解释了图12中的整体贷款额度上升。
4. 仔细分析附录中的数据对比可知，一些资金流较小的生物医药企业在新型冠状病毒之后能贷较大额度的金额。而遭受影响最大的个体经营者，其贷款额度都基本上是在

行业个数

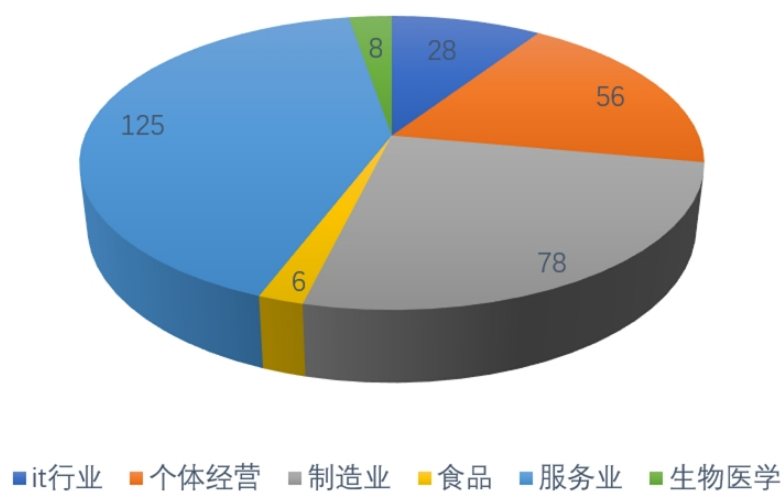


图 11 行业比例图

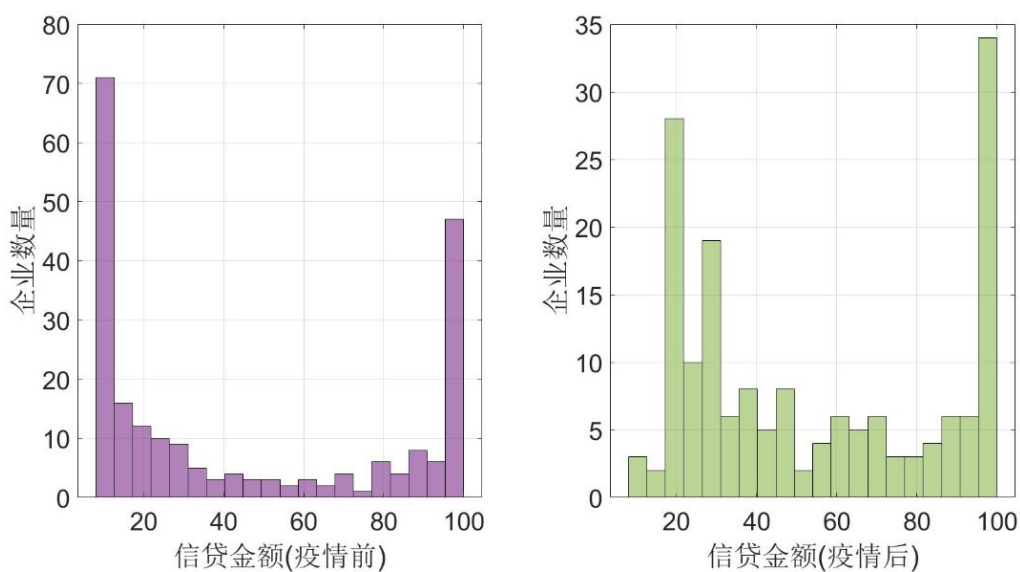


图 12 新型冠状病毒前后信贷金额对比分布直方图

以前的基础上减小数额，有甚者不能申请贷款。IT 行业在此情况下几乎无任何贷款额度变化。从中得出企业受影响程度生物医药 > 食品 > IT 行业 > 服务业 > 制造业 > 个体经营者。

六、模型评价与推广

6.1 优点

1. 运用 TOPSIS 评分模型，从多个数据特征维度对企业信用进行挖掘，避免了相似变量对评估结果的重复贡献。同时 TOPSIS 模型可将所有企业情况纳入进行排序，
2. 利用特征数据，使用决策树与 K-Means 算法对避免手工提取特征，可提高模型鲁棒性。
3. 建立 KMV 信用风险变化评估模型：将突发因素对不同企业的影响转化为对股市相关行业指数的影响，验证其合理性，并用其对信誉评分进行修正。
4. TOPSIS 评分模型与 KMV 信用风险变化评估模型相互照应，考虑了历史数据与即时变动因素，且定量地分析了突发事件前后信用评分的变化，较为客观地评估了突发因素对不同行业带来的影响。

6.2 缺点

1. 由于系统误差与数据局限性，预测值精度与真实值存在一定误差。
2. 数据特征进行处理后所采取的指标仍存在一定局限性。
3. 上下游企业影响定量分析不够全面。

6.3 推广

1. 此模型中运用了遗传算法，在计算过程中，计算时间过长效率较低，应该构建一个更有效的公式找出各个企业的贷款。
2. TOPSIS 评分只适用于确定大量数据内容确定，而不能推广于单个企业数据内容确定，因此改进为一个适用于单个企业数据以及大量企业数据的评价方法。
3. 风险最小与银行收益最大可使用基于帕累托最优的多目标遗传算法进行求解。由于帕累托最优不依赖于权重，优化结果更为客观合理。

参考文献

- [1] 林浩楠. 基于多目标遗传算法和深度学习的投资分配模型研究与应用 [D]. 华南理工大学,2017.
- [2] 谢一飞. 基于改进的 KMV 模型在我国商业银行中的应用研究 [D]. 山东大学,2020.
- [3] 杨晓勇. 商业银行信贷风险量化模型研究 [D]. 西南交通大学,2004.
- [4] 张强. 小额贷款公司贷款结构与经营绩效 [D]. 山东大学,2018.

- [5] 罗亚丽. 地方商业银行客户行业信用风险管理及研究—以曲靖市商业银行为例 [D]. 云南财经大学,2019.58.
- [6] 严浩. 基于 KMV 模型的国内绿色债券风险测度研究 [D]. 西南财经大学,2018.
- [7] Guotai C, Zhichong Z, Abedin M Z. Credit risk rating system of small enterprises based on the index importance[J]. *International Journal of Security and Its Applications*, 2017, 11(6): 35-52.
- [8] Chai N, Wu B, Yang W, et al. A multicriteria approach for modeling small enterprise credit rating: evidence from China[J]. *Emerging Markets Finance and Trade*, 2019, 55(11): 2523-2543.
- [9] Caruso G, Gattone S A, Fortuna F, et al. Cluster Analysis for mixed data: An application to credit risk evaluation[J]. *Socio-Economic Planning Sciences*, 2020: 100850.
- [10] 黄宇其. 我国商业银行中小企业贷款信用风险量化研究 [D]. 上海交通大学,2014.40.
- [11] 黄宇其. 我国商业银行中小企业贷款信用风险量化研究 [D]. 上海交通大学,2014.40.
- [12] Altavilla C, Boucinha M, Peydró J L, et al. Banking supervision, monetary policy and risk-taking: Big data evidence from 15 credit registers[J]. 2020.

附录 A 数据处理与熵权法—matlab 源程序

```
%%TOPSIS主程序
clear;clc
load shuju4.mat
X=shuju4;
%% 第二步：判断是否需要正向化
[n,m] = size(X);
disp(['共有' num2str(n) '个评价对象，' num2str(m) '个评价指标'])
Judge = input(['这' num2str(m) '个指标是否需要经过正向化处理，需要请输入1，不需要输入0: ']);

if Judge == 1
Position =
    input('请输入需要正向化处理的指标所在的列，例如第2、3、6三列需要处理，那么你需要输入[2,3,6]:
    '); % [2,3,4]
disp('请输入需要处理的这些列的指标类型（1: 极小型， 2: 中间型， 3: 区间型）')
Type = input('例如：第2列是极小型，第3列是区间型，第6列是中间型，就输入[1,3,2]: ');

for i = 1 : size(Position,2)
X(:,Position(i)) = Positivization(X(:,Position(i)),Type(i),Position(i));
end
disp('正向化后的矩阵 X = ')
disp(X)
end

%% 第三步：对正向化后的矩阵进行标准化
Z = X ./ repmat(sum(X.*X).^0.5, n, 1);
disp('标准化矩阵 Z = ')
disp(Z)

%% 让用户判断是否需要增加权重
disp("请输入是否需要增加权重向量，需要输入1，不需要输入0")
Judge = input('请输入是否需要增加权重: ');
if Judge == 1
Judge = input('使用熵权法确定权重请输入1，否则输入0: ');
if Judge == 1
if sum(sum(Z<0)) >0
disp('原来标准化得到的Z矩阵中存在负数，所以需要重新对X重新标准化')
for i = 1:n
for j = 1:m
Z(i,j) = [X(i,j) - min(X(:,j))] / [max(X(:,j)) - min(X(:,j))];
end
end
end
```

```

disp('X重新进行标准化得到的标准化矩阵Z为：')
disp(Z)
end
weight = Entropy_Method(Z);
disp('熵权法确定的权重为：')
disp(weight)
else
disp(['如果你有3个指标，你就需要输入3个权重，例如它们分别为0.25,0.25,0.5，
    则需要输入[0.25,0.25,0.5]']);
weight = input(['你需要输入' num2str(m) '个权数。' '请以行向量的形式输入这' num2str(m) '个权重：
    ']);
OK = 0; % 用来判断用户的输入格式是否正确
while OK == 0
if abs(sum(weight) -1)<0.000001 && size(weight,1) == 1 && size(weight,2) == m
OK =1;
else
weight = input('你输入的有误，请重新输入权重行向量：');
end
end
end
else
weight = ones(1,m) ./ m ; %如果不需要加权重就默认权重都相同，即都为1/m
end

%% 第四步：计算与最大值的距离和最小值的距离，并算出得分
D_P = sum([(Z - repmat(max(Z),n,1)) .^ 2] .* repmat(weight,n,1) ,2) .^ 0.5; % D+
    与最大值的距离向量
D_N = sum([(Z - repmat(min(Z),n,1)) .^ 2] .* repmat(weight,n,1) ,2) .^ 0.5; % D-
    与最小值的距离向量
S = D_N ./ (D_P+D_N); % 未归一化的得分
S
disp('最后的得分为：')
stand_S = S / sum(S)
[sorted_S,index] = sort(stand_S , 'descend')

%%中间型
function [posit_x] = Inter2Max(x,a,b)
r_x = size(x,1); % row of x
M = max([a-min(x),max(x)-b]);
posit_x = zeros(r_x,1); %zeros函数用法：zeros(3) zeros(3,1) ones(3)
% 初始化posit_x全为0
for i = 1: r_x
if x(i) < a
posit_x(i) = 1-(a-x(i))/M;
elseif x(i) > b
posit_x(i) = 1-(x(i)-b)/M;
end
end

```

```

else
posit_x(i) = 1;
end
end
end
end
%%最小值
function [posit_x] = Min2Max(x)
posit_x = max(x) - x;
end

function [lnp] = mylog(p)
n = length(p); % 向量的长度
lnp = zeros(n,1); % 初始化最后的结果
for i = 1:n % 开始循环
if p(i) == 0 % 如果第i个元素为0
lnp(i) = 0; % 那么返回的第i个结果也为0
else
lnp(i) = log(p(i));
end
end
end
end

function [posit_x] = Positivization(x,type,i)
if type == 1 %极小型
disp(['第' num2str(i) '列是极小型，正在正向化'] )
posit_x = Min2Max(x);
disp(['第' num2str(i) '列极小型正向化处理完成'] )
disp('-----分界线-----')
elseif type == 2 %中间型
disp(['第' num2str(i) '列是中间型'] )
best = input('请输入最佳的那一个值: ');
posit_x = Mid2Max(x,best);
disp(['第' num2str(i) '列中间型正向化处理完成'] )
disp('-----分界线-----')
elseif type == 3 %区间型
disp(['第' num2str(i) '列是区间型'] )
a = input('请输入区间的下界: ');
b = input('请输入区间的上界: ');
posit_x = Inter2Max(x,a,b);
disp(['第' num2str(i) '列区间型正向化处理完成'] )
disp('-----分界线-----')
else
disp('没有这种类型的指标，请检查Type向量中是否有除了1、2、3之外的其他值')
end
end

function [W] = Entropy_Method(Z)

```



```

%% 计算熵权
[n,m] = size(Z);
D = zeros(1,m); % 初始化保存信息效用值的行向量
for i = 1:m
    x = Z(:,i); % 取出第i列的指标
    p = x / sum(x);

    e = -sum(p .* mylog(p)) / log(n); % 计算信息熵
    D(i) = 1- e; % 计算信息效用值
end
W = D ./ sum(D); % 将信息效用值归一化，得到权重
End

%%数据处理
clc;clear all
x=0.04:0.001:0.15;
a1=[0.8852,0.851,0.8539 ];
b1=[14.27, 13.92,13.67];
c1=[-0.4492,-0.4498,-0.4538];
%     a1 =     0.8852 ;
%     b1 =     14.27 ;
%     c1 =    -0.4492 ;
% a1 =     0.851 ;
% b1 =     13.92 ;
% c1 =    -0.4498 ;
% a1 =     0.8539 ;
% b1 =     13.67 ;
% c1 =    -0.4538 ;
for i=1:3
    subplot(1,3,i)
    subplot(1,3,i)
    y=(1-a1(i)*sin(b1(i).*x+c1(i))).*x;
    p=find(y==max(y));
    plot(x,y,'r','linewidth',2)
    grid on;
    %text(x(p),y(p),'o','color','g')
    text(x(p),y(p),['(',num2str(x(p)),',',',num2str(y(p)),')'],'color','b');
    xlabel('利率')
    ylabel('流失率')
    hold on
end

clc
clear
gongshi=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷

```

```

记录企业的相关数据.xlsx','进项发票信息','a2:a395176');
fapiao=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷

记录企业的相关数据.xlsx','进项发票信息','h2:h395176');

shuzu=zeros(2,b);
for t=1:123
for i=1:210947
if(gongshi(i)==t+123)
if(fapiao(i)==1)
shuzu(1,t)=shuzu(1,t)+1;
end
if(fapiao(i)==0)
shuzu(2,t)=shuzu(2,t)+1;
end
end
end
end

lc
clear
gongshi=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷

记录企业的相关数据.xlsx','进项发票信息','a2:a395176');
fapiao1=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷

记录企业的相关数据.xlsx','进项发票信息','e2:e395176');
fapiao2=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷

记录企业的相关数据.xlsx','进项发票信息','f2:f395176');
fapiao3=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷

记录企业的相关数据.xlsx','进项发票信息','g2:g395176');
fapiao4=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷

记录企业的相关数据.xlsx','进项发票信息','h2:h395176');
b=425-124+1;
shuzu=zeros(3,b);
for t=1:b
for i=1:395175
if(gongshi(i)==t+123)
if(fapiao4(i)==1)
shuzu(1,t)=shuzu(1,t)+fapiao1(i);
shuzu(2,t)=shuzu(1,t)+fapiao2(i);
shuzu(3,t)=shuzu(1,t)+fapiao3(i);
end
end
end

```

```

end
end
A=shuzu';

clc
clear
gongshi=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷
记录企业的相关数据.xlsx','进项发票信息','a2:a395176');
fapiao=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷
记录企业的相关数据.xlsx','进项发票信息','g2:g395176');
fapiao4=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷
记录企业的相关数据.xlsx','进项发票信息','h2:h395176');
b=425-124+1;
shuzu=zeros(1,b);
for t=1:b
for i=1:395175
if(gongshi(i)==t+123)
if(fapiao4(i)==1)
if(fapiao(i)>shuzu(1,t))
shuzu(t)=fapiao(i);
end
end
end

end
end
A=shuzu'

clc
clear
gongshi=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷记录企业的相关数据(最终
版数据).xlsx','销项发票信息','a2:a330806');
fapiao=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷记录企业的相关数据(最终
版数据).xlsx','销项发票信息','v2:v330806');
fapiao4=xlsread('C:\Users\10344\Desktop\2020国家数学建模大赛\C\附件2: 302家无信贷记录企业的相关数据(最终
版数据).xlsx','销项发票信息','h2:h330806');
fapiao5=fapiao4;
t=302;
shuzu=zeros(2,302);

```

```

%% 预处理
for t=1:302
for i=1:330805
if(gongshi(i)==t+123)
if(fapiao4(i)==1)
shuzu(2,t)=fapiao(i);
fapiao4(i)=0;
end
end
end
end

for t=1:302
for i=1:330805
if(gongshi(i)==t+123)
if(fapiao5(i)==1)
if(fapiao(i)>shuzu(1,t))
shuzu(1,t)=fapiao(i);
end
if(fapiao(i)<shuzu(2,t))
shuzu(2,t)=fapiao(i);
end
end
end

end
end
A=shuzu'

```

附录 B KMV 算法—matlab 源程序

```

%%KMV
r=0.0435;
T=1;
EquityTheta = 0.3437794277903090;
EquityTheta = 9.81;
E = 204925.0562835000;
E = 1954672745;
D = 1956776.0000000000;
D = 65326191;
[Va,AssetTheta] = KMVOptSearch(E,D,r,T,EquityTheta);

%%
DPT = D;

```

```

DD = (Va - DPT)/(Va * AssetTheta);
DEF = 1 - normpdf(DD);

function [Va,AssetTheta] = KMVOptSearch(E,D,r,T,EquityTheta)
EtoD=E./D;
x0 = [1,1];
VaThetaX=fsolve(@(x)KMVfun(EtoD,r,T,EquityTheta,x),x0);
Va=VaThetaX(1).*E;
AssetTheta=VaThetaX(2);
End

function F=KMVfun(EtoD,r,T,EquityTheta,x)
d1=(log(x(1).*EtoD)+(r+0.5.*x(2).^2).*T)/(x(2).*sqrt(T));
d2=d1-x(2).*sqrt(T);
F=[x(1).*normcdf(d1)-exp(-r*T).*normcdf(d2)./EtoD-1;
normcdf(d1)*x(1)*x(2)-EquityTheta];
end

```

附录 C 数据处理—Python 源代码

```

import pandas as pd
import numpy as np
data1 = pd.read_excel(r'C:\Users\MSI-NB\Desktop\C\附件2: 302家无信贷记录企业的相关数据.xlsx',
sheet_name='销项发票信息')
data2 =pd.read_excel(r'C:\Users\MSI-NB\Desktop\C\附件2: 302家无信贷记录企业的相关数据.xlsx',
sheet_name='进项发票信息')

s1=data1.groupby('企业代号').sum().rename(columns = {"价税合计": "销售收入"})
s2=data2.groupby('企业代号').sum().rename(columns = {"价税合计": "采购支出"})

s=pd.merge(s1,s2,on=['企业代号'],how='inner')
s['毛收入']=s['销售收入']-s['采购支出']
s['毛利率']=s['毛收入']/s['销售收入']

s.sort_values(by='毛收入',axis = 0,ascending = False)
s.to_excel(r'C:\Users\MSI-NB\Desktop\C\shuju2.xlsx')
from sklearn import tree #导入需要的模块
clf = tree.DecisionTreeClassifier()
clf = clf.fit(train_data,train_labels)
with open("iris.dot", 'w') as f:
f = tree.export_graphviz(clf, out_file=f)
import pydotplus
dot_data = tree.export_graphviz(clf, out_file=None)

```

```
graph = pydotplus.graph_from_dot_data(dot_data)
graph.write_pdf("resultt.pdf")
train_if = pd.DataFrame(ttest, columns=["predict"])
train_if.to_excel(r'C:\Users\MSI-NB\Desktop\C\predict.xlsx')
import seaborn as sns
import matplotlib.pyplot as plt
def test(df):
dfData = df.corr()
plt.subplots(figsize=(9, 9)) # 设置画面大小
sns.heatmap(dfData, annot=True, vmax=1, square=True, cmap="Blues")
plt.savefig('./BluesStateRelation.png')
plt.show()
test(data)
```

附录 D 支撑材料文件列表

- 结果数据.xlsx
- shuju.xlsx
- shuju2.xlsx
- shujvcaozuo2.m
- shujvchazhao.m
- shujvchulimax.m
- shujvnianfen.m
- ceshi.m
- code_Monte_Carlo.m
- createFit.m
- Entropy_Method.m
- fitgaizhen.m
- Inter2Max.m
- KMVcompute.m
- KMVfun.m
- KMVOptSearch.m
- Min2Max.m
- mylog.m
- nihe.m
- Positivization.m
- sjcl.m
- topsis.m