

基于熵权改进的 Topsis 信贷风险评估模型

摘 要

中小企业是企业的重要组成部分，但缺乏启动资金使得企业需要向银行申请贷款，同时银行向中小企业放贷也需要承担一定的风险。本文依照实际情形，建立企业风险评估模型和企业抗风险能力评估模型，根据企业得分情况制定贷款策略，帮助银行有依据性的放贷，减小银行可能的损失。

对于问题一：对题意进行分析后得知，信贷策略由贷款额度、贷款期限、贷款利率组成，而是否放贷以及贷款额度由企业的信贷风险决定。查阅相关文献再结合附件一中的数据，确定了信贷风险的评价指标为：信誉评级，是否违约，企业净利润，企业交税金额，企业收入方差。利用 Python 对数据进行预处理，剔除无效发票和评级为 D 的企业，计算出指标数据后，本文使用熵权法改进的 Topsis 法，对每家企业的信贷风险进行定量打分。根据打分结果截取排名前 94% 的企业发放贷款，依照比例确定贷款额度。按照信誉评级对企业进行分类，使用网格搜索法计算出银行收益最大时不同信誉评级的利率情况，最终确定出信贷策略，结果如下表所示：

| 信誉评级 | 贷款期限 | 对应利率 | 银行收益 |
|------|------|--------|----------------|
| A | 1 年 | 0.0465 | 0.04065 |
| B | 1 年 | 0.0585 | |
| C | 1 年 | 0.0585 | |

对于问题二：由于附件 2 中的 302 家企业并没有相应的信贷记录，所以银行要根据企业的已知信息来判断其信誉等级和信贷风险。在问题一的指标基础之上，提取出了进项总金额、销项总金额这两个指标、去除了是否违约的指标，使用 BP 神经网络，预测出 302 家企业的信誉等级，之后再使用模型一的风险评估模式计算出相应的得分，在结合附件三，最终计算得出不同企业的贷款政策方法，如下表所示：

| 信誉等级 | 贷款期限 | 贷款利率 | 银行收益(万元) |
|------|------|--------|-----------------|
| A | 1 年 | 0.0465 | 386.3957 |
| B | 1 年 | 0.0585 | |
| C | 1 年 | 0.0585 | |

对于问题三：对题意进行分析，问题三在问题二的基础上增加了未知因素对企业的影响，未知因素无法量化分析但企业自身的抗风险能力却可以通过数据反应，因此本文采用模糊综合评价法，以企业自身的管理和经营能力入手，确定企业抗风险能力的指标为负数和作废发票的金额，将企业按照行业划分，进而建立起整个行业的抗风险能力评估模型。12 个行业抗风险能力(越大抗风险能力越强)结果如下所示：

| 行业名称 | 零售业 | 工业 | 手工业 | 制造业 | 种植业 | 服务业 |
|-------|-------|-------|-------|-------|-------|-------|
| 抗风险能力 | 0.50 | 0.46 | 0.458 | 0.383 | 0.361 | 0.336 |
| 行业名称 | 医疗业 | 其他 | 物流业 | 互联网 | 房地产 | 餐饮业 |
| 抗风险能力 | 0.325 | 0.289 | 0.278 | 0.275 | 0.244 | 0.238 |

将得出的抗风险能力作为新的评价指标带入问题一建立的信贷风险评价模型中，计算出最大收益下的信款策略。

关键词： 熵权法 Topsis 法 BP 神经网络 模糊综合评价 数据处理

一、问题重述

1.1 问题背景：

在实际生活中，由于中小企业规模较小，也缺少抵押资产，因此银行根据信贷政策、企业交易票据和对上下游企业的影响，向实力强、供求关系稳定的企业提供贷款。并根据中小企业的实力和信誉对其信贷风险做出评价，根据信贷风险等因素来确定是否放贷以及贷款额度、利率个期限等信贷策略。

1.2 问题一重述：

问题一要求对附件 1 总的 123 家企业的信贷风险进行量化分析，并根据分析数据来给出该银行在年度信贷总额固定时对这些企业的信贷策略。

1.3 问题二重述：

问题二要求在问题一的基础上，对附件 2 中的 302 家企业的信贷风险也进行量化分析，并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。

1.4 问题三重述：

在企业生产经营中可能会受到一些突发情况的影响，并且不同的突发因素对不同行业，不同企业会有不同的影响。综合考虑附件 2 中各企业的信贷风险和可能的突发因素对各企业的影响，给出该银行在年度信贷总额为 1 亿元时的信贷调整策略。

二、问题分析

2.1 问题一的分析

问题一要求对附件一中 123 家企业的信贷风险进行量化评估，根据评估结果确定银行在年度信贷总额固定的条件下，对这些企业的信贷策略。根据题意可知，银行首先根据企业的实力、供求关系、信誉对其进行信贷风险评估，依据信贷风险等相关因素从而确定是否给予企业贷款，并对信誉高且信贷风险小的企业给予利率优惠，最终形成信贷策略。查阅相关文献得知，可由以下五个指标评估每个企业的信贷风险。

评价指标：信誉评级，是否违约，企业净利润，企业交税金额，企业收入增长率。确定评价指标后，本文使用熵权法改进后的 Topsis 法对这五个评价指标进行处理，通过熵权法计算每个指标的权重，最终建立企业信贷风险评估模型，依据模型计算出的信贷风险得分情况确定发放贷款的企业，并按得分比例确定企业可以贷款的金额。将这些企业按信誉等级进行划分，统计出每种等级下企业贷款的总金额，依照不同等级给予企业不同利率，参考附录三中的客户流失率，使用网格搜索法进而计算出每个利率下银行获利的总金额。找出银行最大获利时，每种信誉所给的利率，最终确定信贷策略。

2.2 问题二的分析

问题二要求在问题一的基础上，对附件二中 302 家企业的信贷风险进行量化分析，在对附件二中的数据进行分析后发现，由于这些企业并没有信贷记录，所以信誉等级和是否违约并没有相应的数据，因此本文使用 BP 神经网络，根据这些企业已有的数据，提取出了企业三年的进项总金额，销项总金额，净利润，交税金额，月利润方差，这 5 个指标，将附件一中的 123 家企业按 70%、15%、15%作为训练集、验证集、测试集，并选择 Levenberg-Marquardt 方法，训练出对信誉等级的预测模型，再带入附件

二中的数据，预估出 302 企业的信誉等级，最后使用问题一中的模型，以 302 家企业的净利润、税收、月利润方差、信誉等级作为指标，计算得出相应的分数，根据这些得分，使用问题一中的方法，得出收益最大化的信贷策略。

2.3 问题三的分析

问题三在问题二的基础上增加了未知因素对企业的影响，企业受到的影响主要由两方面决定，一方面是外界的影响大小，一方面是企业自身的抗风险能力。外界的影响无法预知，但企业自身的抗风险能力却可以通过企业的信息反应出来，在查阅相关文献并结合附件二中的发票信息，可以确定了企业抗风险能力的指标，依照指标建立企业抗风险能力的模型，未知因素的影响不仅会影响到企业自身，而且会影响到企业所处的整个行业，因此本文对 302 家企业的行业进行划分，使用模糊综合评价按照行业建立抗风险能力模型，在解出抗风险能力的数值后，将其作为新的评价指标代入问题一所建立的信贷风险评估模型中。利用得出的风险分数重新对银行贷款策略进行调整，得出银行在利益最大化的情况下的贷款策略。

三、模型假设

1. 假设企业提供的发票全部真实有效。
2. 假设企业不会在贷款期限内发生重大变故。
3. 假设突发因素对各个行业都有一定影响。

四、符号说明

| 符号 | 说明 |
|---------------|---------------------|
| X | Topsis 中的评价矩阵 |
| Z | 标准化后的矩阵 |
| Y_j | 隐层单元 |
| Y_k | 表示输出层第 k 个节点的实际输出 |
| $f(x)$ | 激活函数 |
| W_{ji} | 输入层到隐层之间的连接权向量 |
| W_{kj} | 隐层到输出层之间的连接权向量 |
| ϑ_j | 隐层阈值 |
| γ_k | 输出层的阈值 |
| O_{pk} | 第 P 个输入模式的期望输入 |
| Y_{pk} | 网络的实际计算输出 |
| U | 因素集 |
| A | 权重集 |
| V | 评语集 |
| R | 模糊综合判断矩阵 |

五、模型的建立与求解

5.1 问题一模型的建立与求解

问题一本文采用 Topsis 法通过五个指标（信誉评级，是否违约，企业净利润，企业交税金额，企业收入增长率）并使用熵权法对每一个指标赋权，得到企业的信誉得分（分值越高风险越低）。依据崔建荣^[1]对银行面向中小企业信贷满足率的研究，本文划分得分排名前 94% 的企业为发放贷款企业，依据这些企业的得分占比划分贷款金额。得到不同信誉等级下企业贷款的总金额，使用网格搜索法将不同信誉等级的利率以及对应的流失率带入利润方程中，最终得出银行获得最大利润下的信贷策略。

5.1.1 评价指标的确定

分析题意得知，银行依据企业的信贷风险向企业提供贷款，并对信誉高、信贷风险小的企业给予利率优惠，同时银行依据企业实力、供求关系稳定性、企业信誉对企业的信贷风险做出评估，进而确定贷款的额度以及利率，贷款期限都为 1 年，最终形成信贷策略。对信贷策略进行分析后，可得如下思维导图：

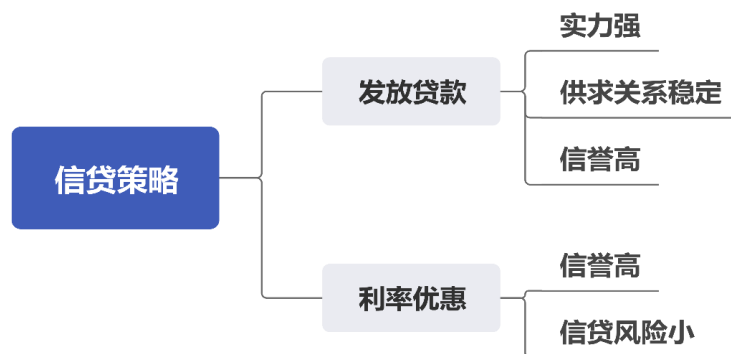


图 1 信贷策略思维导图

根据图 1 中的信息，初步确定了银行对企业信贷风险的评价指标为：企业的实力，企业供求关系的稳定性以及企业的信誉。根据马刚宇^[2]对企业高质量发展的指标分析评价体系构建的研究，文中通过使用杜邦分析法论证了公司高质量发展的评价指标主要有财务净收入，资产负债率，企业交税额，员工收入状况。题目要求对企业的信贷风险进行量化的评估，因此结合附件一所给出的 123 家企业的发票数据和信誉状况，本文确定了企业实力的评价指标为：企业净利润、交税金额，企业供求关系稳定性的评价指标为：企业月度净利润的方差，企业信誉的评价指标为：企业信誉的评价指标为：信誉评级、是否违约。依照上述分析可做出信贷风险指标导图：

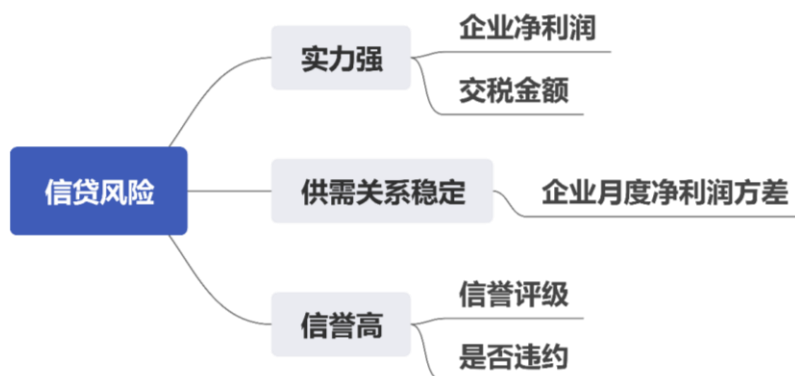


图 2 信贷风险思维导图

依据图 2 信息，最终确定了信贷风险的评价指标为：企业净利润、交税金额、企业月度净利润方差、信用评级、是否违约。

5.1.2 数据预处理

为了对企业的信贷风险进行定量的分析，首先先将附录一中的无效发票和信誉评价为 D 的企业剔除，然后依照发票信息对每一个企业的净利润 W ，交税金额 M ，企业月度净利润方差 Z 进行计算，计算公式如下所示。

净利润 W ：

$$W = W_{\text{销项金额}} - W_{\text{进项价税合计}} \quad (1)$$

交税金额 M ：

$$M = \sum_{i=1}^n M_i \quad (2)$$

企业月度净利润方差 C ：

$$C = \frac{\sum_{i=1}^n (W_i - \bar{W})^2}{n} \quad (3)$$

计算完成后，企业代号 E20—E30 的结果如下表所示(所有企业详见附件 1)：

表一 企业量化实力表

| 企业代号 | 净利润(万元) | 交税金额(万元) | 月度利润方差(万元) |
|------|----------|----------|------------|
| E20 | -1251.34 | 3029.56 | 6.78033E+8 |
| E21 | -478.18 | 1130.99 | 8.18074E+8 |
| E22 | 3704.38 | 1095.24 | 7.43953E+8 |
| E23 | -1775.58 | 2880.56 | 2.78669E+8 |
| E24 | 5428.96 | 1372.80 | 3.79633E+8 |
| E25 | 2799.28 | 583.67 | 1.45414E+8 |
| E26 | -599.74 | 578.10 | 1.63934E+8 |
| E27 | -483.14 | 460.15 | 3.41627E+7 |
| E28 | 4761.48 | 474.39 | 3.26811E+8 |
| E29 | 4498.48 | 180.49 | 5.02063E+8 |
| E30 | 4727.25 | 661.50 | 5.7363E+8 |

分析表中数据，净利润表示企业近几年的收益情况（正为盈利，负为亏损），交税金额为企业近几年的交税情况，月度方差表示企业发展的平稳性，利润方差小的企业在近几年发展中供求关系较为平稳，根据表中企业利润和交税金额，做出如下图：

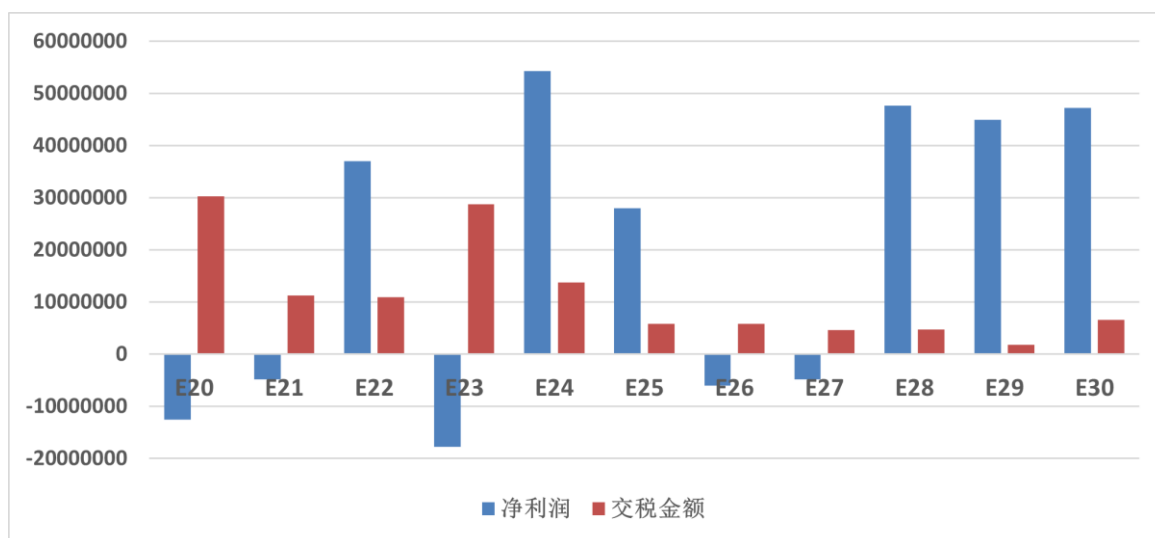


图 3 企业收益及交税情况图

图 3 中，蓝色线条代表企业净利润，红色线条代表企业交税金额，对图 3 进行分析可知，在企业的净利润为正数时（企业正常盈利）企业的净利润越高所缴税额也随之增高，但在企业净利润为负数时（企业亏损）企业的净利润与之所缴税额无直接联系。亏损最严重的两家企业（E20,E23）所缴纳税款反而最多，这可能是企业早期盈利较多，后期由于转型或者衰落所导致的现象。

企业的信誉评级以及违约情况都为定性指标，为了进行定量的评估，因此本文将这些指标用虚拟变量代替，虚拟变量的转化如下所示：

$$\text{信誉评级} \begin{cases} A = 100 \\ B = 75 \\ C = 50 \\ D = 25 \end{cases} \quad \text{是否违约} \begin{cases} \text{是} = 0 \\ \text{否} = 10 \end{cases} \quad (4)$$

收集 123 家企业的信誉评级，违约情况，并将企业分别按信誉评级和违约情况进行划分，得到如下图像：

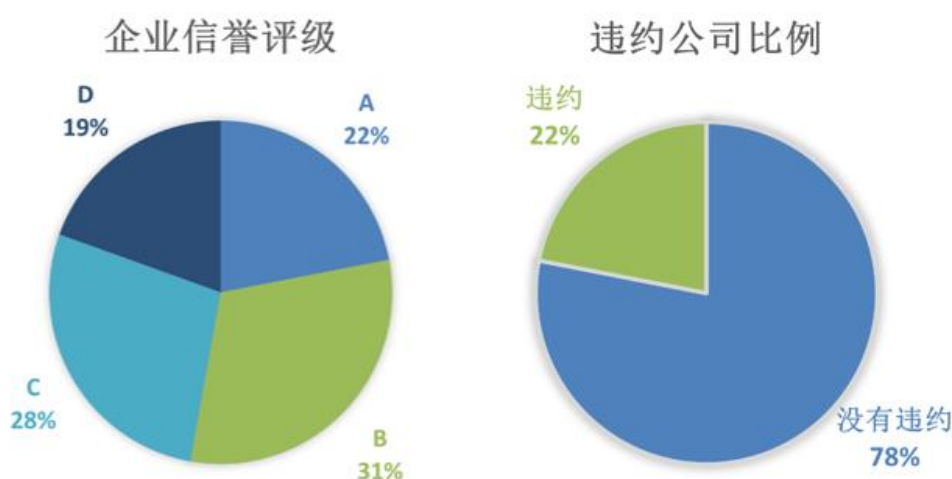


图 4 企业信誉(左)及违约情况(右)图

分析企业信誉比例图(左)可知，信誉评级为 A 的企业占有所有企业的 22%；为 B 的企业占有所有企业的 31%；为 C 的企业占有所有企业的 28%；为 D 的企业占有所有企业的

19%，信誉评级为 B 的企业占比重最大。分析违约公司比例图(右)可知，违约公司占总比例的 22%，而根据附录一中的数据来看，信誉评价为 D 的企业都有违约记录，因此信誉评价为 D 的企业占了违约总数的 86.3%。

5.1.3 Topsis 法建立信贷风险评估模型

在数据预处理完成后，所有评价指标(企业净利润、交税金额、企业月度净利润方差、信誉评级、是否违约)都已经被量化，由于数据量大且决策因素较多其他评价方法无法客观且量化的反应出企业的信贷风险，接下来本文将采用 Topsis 法对所有企业的信贷风险进行打分。Topsis 简略流程图如下图所示：

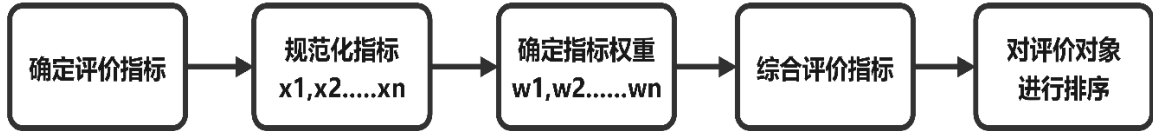


图 5 Topsis 流程图

第一步：建立评价矩阵

现有 123 个待评价的企业，5 个评价指标，因此可建立如下的矩阵：

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & \dots & \dots & x_{25} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & \dots & x_{n5} \end{bmatrix} \quad (5)$$

矩阵共 123 行、5 列，123 行代表评价对象，5 列代表评价指标。

第二步：正向化处理

首先分析指标类型，企业净利润：代表企业的盈利与亏损金额，因此为极大型；交税金额：间接反映企业以前的盈利额，因此为极大型；企业月度净利润方差：反应企业的供需关系是否稳定，因此为中间型；信誉评级：银行内部人员对企业进行的信誉评定，数值越高等级越高，因此为极大型；是否违约：企业的违约记录，违约为 0 否则为 10，因此为极大型。

五个评价指标中，四个已经为极大型指标，只需将企业月度净利润方差转化为极大型的指标即可，转化公式如下所示：

$$M = \max \{|x_i - x_{best}|\}, \bar{x}_i = 1 - \frac{|x_i - x_{best}|}{M} \quad (6)$$

公式 6 中， $\{x_i\}$ 为企业月度净利润方差的序列， x_{best} 为最佳数据也即是 0。

第三步：矩阵标准化

对已经正向化处理的矩阵进行标准化处理，目的是为了消除指标间不同量纲的影响。标准化后的矩阵记为 Z，那么对于 Z 中的每一个元素 z 有如下公式：

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^n x_{ij}^2}} \quad (7)$$

得到标准化后的矩阵 Z：

$$Z = \begin{bmatrix} z_{11} & z_{12} & \dots & z_{15} \\ z_{21} & z_{22} & \dots & z_{25} \\ \dots & \dots & \dots & \dots \\ z_{n1} & z_{n2} & \dots & z_{n5} \end{bmatrix} \quad (8)$$

第四步：熵权法确定指标权重

首先计算概率矩阵，这里记为 P，对于 P 中的每一个元素 p 有如下计算公式：

$$p_{ij} = \frac{\tilde{z}_{ij}}{\sum_{i=1}^n \tilde{z}_{ij}} \quad (9)$$

然后计算每个指标的信息熵，对于第 j 个指标而言，其信息熵 e 的计算公式如下所示：

$$e_j = \frac{1}{\ln n} \sum_{i=1}^n p_{ij} \ln(p_{ij}) \quad (j=1, 2, 3, 4, 5) \quad (10)$$

计算信息效用值的计算公式为 $d_j = 1 - e_j$ ，最后通过信息效用值 d 计算出每个指标的权重 H，计算公式如下所示：

$$H_j = \frac{d_j}{\sum_{i=1}^n d_j} \quad (j=1, 2, 3, 4, 5) \quad (11)$$

第五步：计算得分并归一化处理

定义第 i 个评价对象与最大值的距离为： D_i^+ ，与最小值的距离为 D_i^- ，那么可以计算出第 i 个评价对象未归一化的得分记为 S_i ，其计算公式如下所示：

$$S_i = \frac{D_i^-}{D_i^+ + D_i^-} \quad (12)$$

最后对得分进行归一化处理得出最终得分 F，公式如下：

$$F = \frac{S_i}{\sum_{j=1}^n S_j} \quad (13)$$

5.1.4 风险评估模型的求解

将已经在 excel 中预处理好的数据导入 Matlab 中，使用编写好的 Topsis 算法（详见附录 1）对模型进行求解。解得五个评价指标的权重如下表所示：

表二 指标权重表

| 评价指标 | 企业净利润 | 交税金额 | 企业月度净利润方差 | 信誉评级 | 是否违约 |
|------|--------|--------|-----------|--------|--------|
| 权重 | 0.1876 | 0.0120 | 0.0048 | 0.7895 | 0.0059 |

分析表二可知，由熵权法得出的权重信誉评级和企业净利润权重之和占据了 98% 的权重，使用该权重结果可以得到所有企业归一化后的得分，本文按照崔建荣^[1]对银行对中小企业信贷满足率的研究，截取前 94% 给予贷款资格，并按得分所占前 94% 总得分的比例分配贷款比例，排名前十的结果如下表所示(详细结果见附录 1)：

表三 企业排名情况

| 排名 | 企业代号 | 风险得分 | 信誉评级 | 贷款比例 |
|----|------|--------|------|--------|
| 1 | E1 | 0.0397 | A | 0.0409 |
| 2 | E4 | 0.0182 | C | 0.0187 |
| 3 | E7 | 0.0166 | A | 0.0171 |
| 4 | E9 | 0.0158 | A | 0.0162 |
| 5 | E2 | 0.0155 | A | 0.0159 |
| 6 | E13 | 0.0154 | A | 0.0158 |
| 7 | E6 | 0.0153 | A | 0.0158 |
| 8 | E19 | 0.0153 | A | 0.0158 |
| 9 | E8 | 0.0152 | A | 0.0156 |
| 10 | E24 | 0.0150 | A | 0.0155 |

分析表三数据，排名前十的有 90%为信誉评级 A 级的企业，这与信誉评级的权重较大有关。信誉评级为 C 的 E4 企业排名第二，反映出 E4 企业的企业实力要远超其他企业。

在银行贷款总金额固定的情况下，按各个企业的贷款比例为其分配贷款，给予不同信誉评级的企业不同的利率，考虑到使银行收益最大化，可建立如下公式：

$$W_{\text{银行}} = D_A l_A (1 - p_A) \times D_B l_B (1 - p_B) \times D_C l_C (1 - p_C) \quad (14)$$

公式 14 中，W 为银行能收到的利润，D 表示一个信誉评级的总贷款金额，l 表示贷款利率，p 为一定利率下的流失率。通过网格搜索法计算出银行利润 W 的最大值，记录下此时的利率情况(python 代码详见附录 3)，贷款策略结果如下表所示：

表四 银行最佳决策

| 信誉评级 | 贷款期限 | 对应利率 | 银行收益 |
|------|------|--------|----------------|
| A | 1 年 | 0.0465 | 0.04065 |
| B | 1 年 | 0.0585 | |
| C | 1 年 | 0.0585 | |

由上表可见，在银行年贷款额限定的情况下，银行对信誉为 A 的企业给予 0.0465 的利率，对信誉为 B 的企业给予 0.0585 的利率，对信誉为 C 的企业给予 0.0585 的利率，此时银行的收益最大，最大值为 0.04065。

5.2 问题二模型的建立与求解

5.2.1 BP 神经网络预测模型的建立

由于附件二中 302 家企业并没有信誉等级，因此为了对其信贷风险进行量化分析，需要预估出相应的信誉等级。我们综合比较了多分类线性判别分析和 bp 神经网络，选择了 BP 神经网络，多分类线性判别分析测试如下表所示：

表五 多分类线性判别预测结果表

| 实测 | 预测 | | | 正确百分比 |
|-------|-------|-------|-------|-------|
| | 1 | 2 | 3 | |
| 1 | 21 | 0 | 0 | 100% |
| 2 | 0 | 20 | 0 | 100% |
| 3 | 0 | 0 | 17 | 100% |
| 总体百分比 | 36.2% | 34.5% | 29.3% | 100% |

因为在使用多分类线性判别分析时,将数据分为训练组和测试组,用训练组来训练出模型,测试组的数据测试模型,结果发现训练组的回归结果十分准确(如表五所示),而测试组的预测准确率为 0%,因此推测模型出现了过拟合现象,放弃使用多分类线性判别分析。

BP 神经网络^[3]是在人类对大脑神经网络认识理解的基础上构建的某种数学模型,它由大量简单原件相互连接而成,具有高度的并行性、良好的容错性和联想记忆功能,十分强大的自适应、学习能力。

在人工神经网络的实际应用中,BP 网络广泛应用于函数逼近、模式识别与分类、数据压缩等,80~90%的人工神经网络模型是采用 BP 网络或它的变化形式,它是前馈神经网络的核心部分,体现了人工神经网络最精华的部分。

由于这些突出的优点,结合企业的净利润、税收、月利润方差和信誉等级之间存在非线性问题,所以本文认为 BP 神经网络在分析四个指标之间的关系中的应用是可行的。

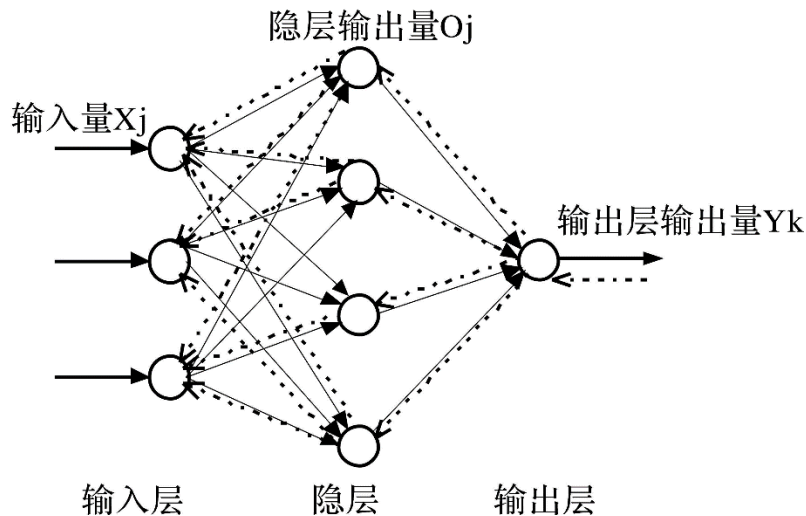


图 6 神经网络模型图

BP 神经网络是一种单向传播的多层前馈神经网络,分为输入层、隐含层和输出层(如图 6 所示),层与层之间多采用全连接方式,同一层单元之间不存在相互连接。其中隐含层可为多层。数学上已经证明过 3 层网络即可模拟任何非线性问题,BP 神经网络的学习过程有正向传播和反向传播两部分组成,在正向传播过程中,从输入层经过隐层神经元激活后,传向输出层,每一层神经元的状态只影响下一层神经元状态。得到的输出与目标值计算 Loss,并进行反向传播,此时 Loss 从输出层向输入层传播并沿途调整各层间连接权值和阈值,以使误差不断减小,来提升精度。

$$\text{隐含层: } Y_j = f\left(\sum_{i=1}^l W_{kj} X_i - \theta_j\right) \quad (15)$$

$$\text{输出层: } Y_k = f\left(\sum_{i=1}^l W_{ki} X_i - \gamma_k\right) \quad (16)$$

上式中: X_i 表示网络的输入; Y_j 表示隐层单元和 j 个节点的输出; Y_k 表示输出层第 k 个节点的实际输出; $f(x)$ 表示激活函数; W_{ji} 、 W_{kj} 分别表示输入层到隐层和隐层到输出层之间的连接权向量; ϑ_j 、 γ_k 分别表示隐层和输出层的阈值。

$$\text{定义误差函数为: } E_p = \frac{1}{2} \sum_k (O_{pk} - Y_{pk})^2 \quad (17)$$

式中: O_{pk} 为第 P 个输入模式的期望输入; Y_{pk} 为网络的实际计算输出。

按梯度下降法求权重、节点阈值, 调整值如下:

$$\Delta w_{ij}(t+1) = \eta \delta_j Y_i + \alpha \Delta w_{ij}(t) \quad (18)$$

$$\Delta \theta_{ij}(t+1) = \eta \delta_j + \alpha \Delta \theta_{ij}(t) \quad (19)$$

式中: Y_i 是神经元 i 在 t 时刻的实际输出; $0 < \eta < 1$ 为学习效率; α 为动量因子, δ_j 为节点的误差, 根据节点的形式不同分别计算, 当为隐含层节点时:

$$\delta_j = Y_j(1 - Y_j) \sum_k (\delta_k w_{kj}) \quad (20)$$

为输出节点时:

$$\delta_j = Y_j(1 - Y_j) (O_j - Y_j) \quad (21)$$

式中: Y_i 是神经元 i 在 t 时刻的实际输出; k 为神经元 j 的下一层神经元编号。

BP 学习算法对网络进行训练, 学习步骤如下:

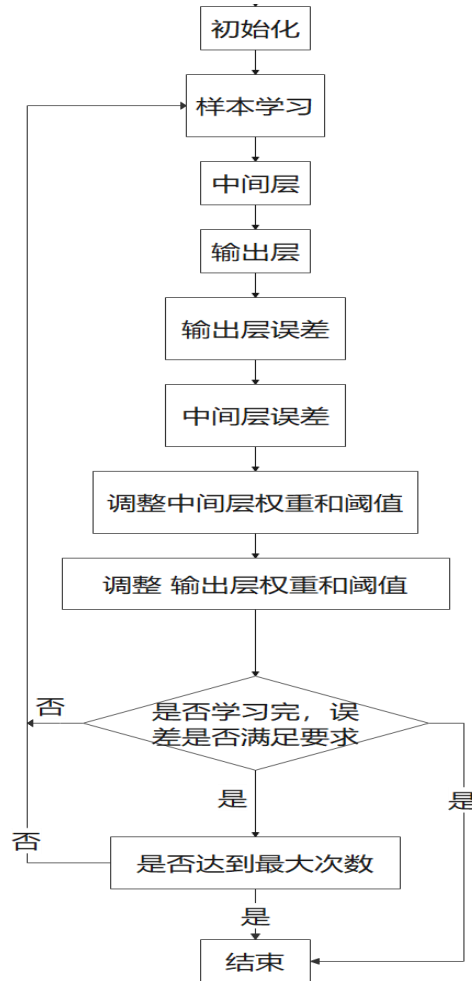


图 7 神经网络算法流程图

本文的 BP 神经网络模型建立步骤如下：

Step1: 输入数据于输出数据的导入。去除附件一中信誉等级为 D 的数据后，选取前 85 个分组用于神经网络的构造。将影响指标作为输入数据，信誉等级作为输出数据导入到工作区中，从而确定了输入、输出维数分别为 5，1。

Step2: 设置输入数据所占比例。为保证模型的可靠性，分别设置训练数据、验证数据、测试数据所占比例分别为 70%、15%、15%。

Step3: 确定隐含层神经元。经过多次尝试，最终选择神经元为 5，因此，本文的 BP 神经网络隐含层有 15 个神经元。

Step4: 神经网络训练。样本期望值输出误差小于给定收敛值时，停止训练，否则继续训练。

5.2.2 模型的求解

根据上述模型，先导入相关数据，使用 MATLAB 建立神经网络模型（代码详见附录 5），输入数据的比例，隐含层神经元个数，经过多次训练得到了如下的训练结果图：

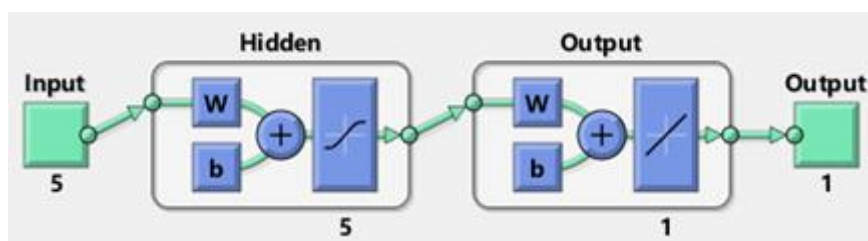


图 8 神经网络图

最终确定了预测信誉等级的神经网络模型为 5-5-1 结构，如图四所示。并且经过多次训练后发现，数据具有明显的收敛迭代效果，迭代速度也较快，本次训练迭代了 22 次后达到了要求，如下图所示：

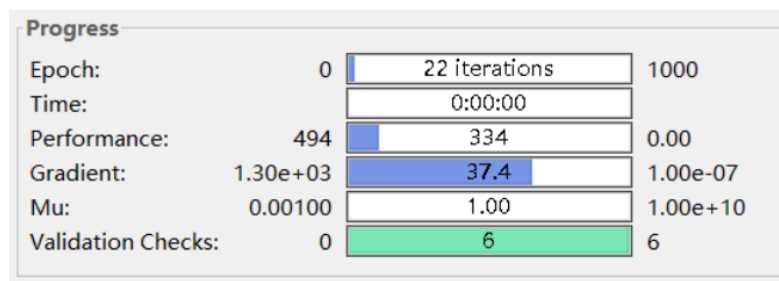


图 9 迭代结果图

一般来说，一个函数的拟合效由 R^2 来确定， R^2 越接近于 1，说明拟合效果越好，从图 5 可知训练数据的 $R=0.74319$ ，实验组的预测准确率在 74% 上下波动，如下表所示：

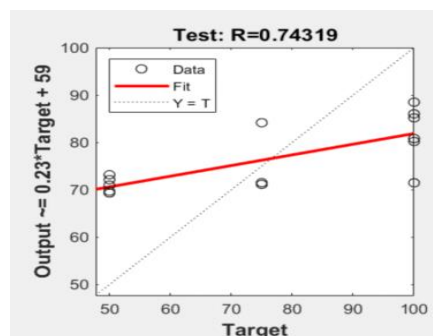


图 10 拟合优度图

在模型训练完成之后，再导入需要预测的数据，使用 Matlab 编程(代码详见附录 5)，用训练好的神经网络模型，对其进行预测得到了信誉分数，根据附件一中每个信誉等级所占的比例(A:27.1% , B:38.2% ,C:34.7%)，将企业划分信誉评级，划分结果如下表所示（详细数据见附录 4）

表六 302 家企业信誉等级表

| 企业代号 | 信誉分数 | 信誉评级 | 企业代号 | 信誉等级 | 信誉评级 |
|------|----------|------|------|----------|------|
| E124 | 73.28669 | A | ... | ... | |
| E125 | 53.63663 | A | E410 | 66.18703 | C |
| E126 | 107.4653 | A | E411 | 66.14415 | C |
| E127 | 87.81984 | A | E412 | 66.17001 | C |
| E128 | 91.27201 | A | E413 | 66.16747 | C |
| E129 | 95.88482 | A | E414 | 66.18425 | C |
| E130 | 78.32368 | A | E415 | 66.13372 | C |
| E131 | 90.35146 | A | E416 | 66.14715 | C |
| E132 | 92.38489 | A | E417 | 66.13369 | C |
| E133 | 78.09453 | A | E418 | 66.1894 | C |
| E134 | 85.91625 | A | E419 | 65.12539 | C |
| E135 | 86.06216 | A | E420 | 65.95595 | C |
| E136 | 77.63985 | A | E421 | 66.15998 | C |
| E137 | 80.02616 | A | E422 | 66.12514 | C |
| E138 | 83.96389 | A | E423 | 66.12727 | C |
| E139 | 95.35519 | A | E424 | 66.11094 | C |
| ... | ... | | E425 | 66.12997 | C |

在得到了相应企业的信誉预估等级之后，在使用第一题中的风险评价模型，对每个企业进行打分，最后选取了前十个得分最高的企业画出了相应的条形图，如下图所示：

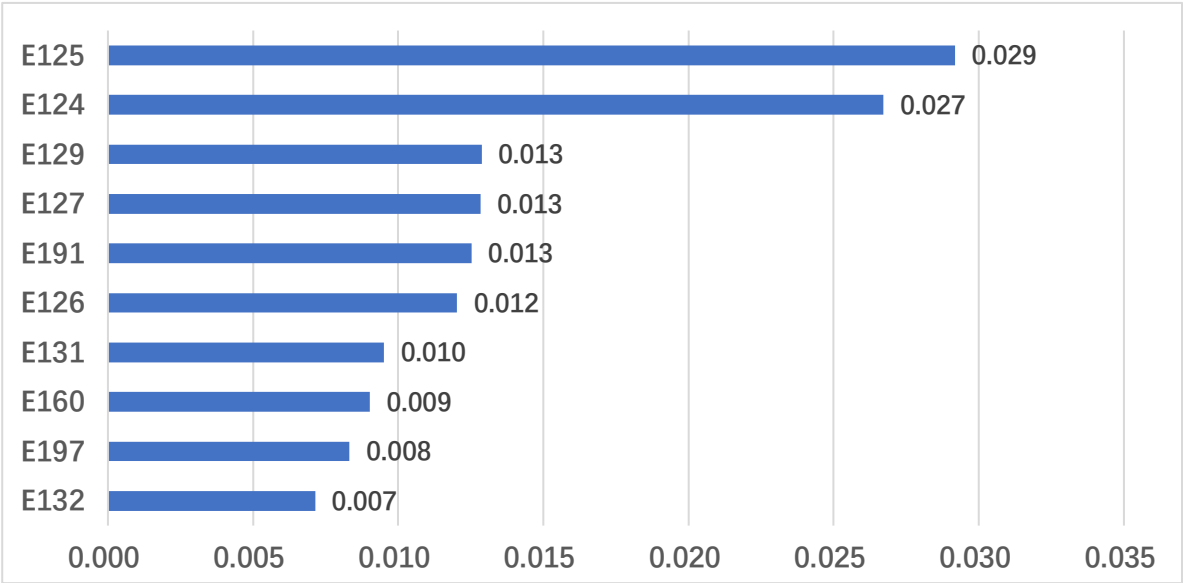


图 11 302 家企业得分前十图

图 6 中，x 轴为企业归一化后的得分情况，y 轴为企业的代号。由上图可知，E125、E129 企业得分较高，实力强，并且供求关系稳定，所以提供的贷款额度比其他企业要

高，E191、E131、等一些企业得分稍差一点，因此贷款的额度也会收到影响。

得到所有企业的得分后，截取得分排名前 94%的企业发放贷款(共 284 家企业)，贷款金额由各企业的得分所占比例决定，由于本题限制银行总贷款金额为 1 亿元，且每个企业的贷款额度区间在 10 万~100 万，由图 6 中可知，前六家企业的贷款占比超过了 0.01，因此将他们的贷款金额限制在 100 万元。结合问题一中的网格搜索法，解出在银行总贷款金额固定的情况下，银行收益最高时各信誉评级所给的利率，并给出贷款策略，贷款策略结果如下表所示：

表七 银行策略表

| 信誉等级 | 贷款期限 | 贷款利率 | 银行收益(万元) |
|------|------|--------|----------|
| A | 1 年 | 0.0465 | 386.3957 |
| B | 1 年 | 0.0585 | |
| C | 1 年 | 0.0585 | |

由上表可确定银行贷款策略，对信誉等级为 A 的企业贷款利率为 0.0465，对信誉等级为 B 的企业贷款利率为 0.0585，对信誉等级为 C 的企业贷款利率为 0.0585,银行预计最高收益为 386.39 万元。

5.3 问题三模型的建立与求解

5.3.1 抗风险能力指标的确立

分析题意后可知，问题三相较问题二的基础上需要考虑企业未来可能遭受的突发因素对企业的影响，突发因素往往对不同行业，不同类别的企业有不同的影响。因此本文以行业类别对 302 家企业进行划分，使用 python 对所有企业名称按关键字进行筛选(python 代码详见附录 6)，所有的行业类型及其中部分企业的划分结果如下所示(详细结果见附录 7)：

表八 行业类型表

| 行业类型 | 部分企业代号 | 部分企业名称 | 企业个数 |
|-------|--------|---------------|------|
| 种植行业 | E179 | ***园林有限公司 | 6 |
| 物流行业 | E142 | ***运业有限公司 | 12 |
| 零售行业 | E124 | 个体经营 E124 | 56 |
| 房地产行业 | E135 | ***建设工程有限公司 | 74 |
| 服务行业 | E148 | ***汽车销售服务有限公司 | 52 |
| 制作业 | E136 | ***机械有限责任公司 | 23 |
| 手工业 | E196 | ***地质制图印刷厂 | 16 |
| 餐饮行业 | E141 | ***食品有限公司 | 5 |
| 互联网行业 | E143 | ***电子科技有限公司 | 35 |
| 医疗行业 | E195 | ***医药有限公司 | 4 |
| 工业 | E180 | ***电气有限公司 | 3 |
| 其他行业 | E167 | ***体育文化股份有限公司 | 16 |

分析表八可知，302 家企业共划分出 12 个不同类型的行业，其中房地产行业申请贷款的比例占申请贷款企业数量的 24.5%，可能时由于房地产企业前期投入资金较大导致的。而工业的占比最小，仅有 1%，可能是由于工业的企业数量较少，或是工业企业的经营情况稳定。

根据周晓丹^[4]在‘外向型中小企业抗风险能力评价研究’一文中的理论，企业抗风险能力由外部环境和内部因素两方面组成，并通过企业的外贸能力、管理能力、创新能力、技术与生产能力体现。结合附件二中的发票数据，本文以企业的销项进项作废

发票金额反应企业的管理能力，以进项的负数发票金额和、月度盈利方差反应企业的外贸能力。通过对这四个指标按行业计算平均值，反应出行业总体的抗风险水平，因此行业抗风险能力的指标为：销项作废发票金额平均值、进项作废发票金额平均值、负数发票金额平均值、月度盈利方差平均值。利用计算出的结果可做出如下图像（详细结果见附录 8）：

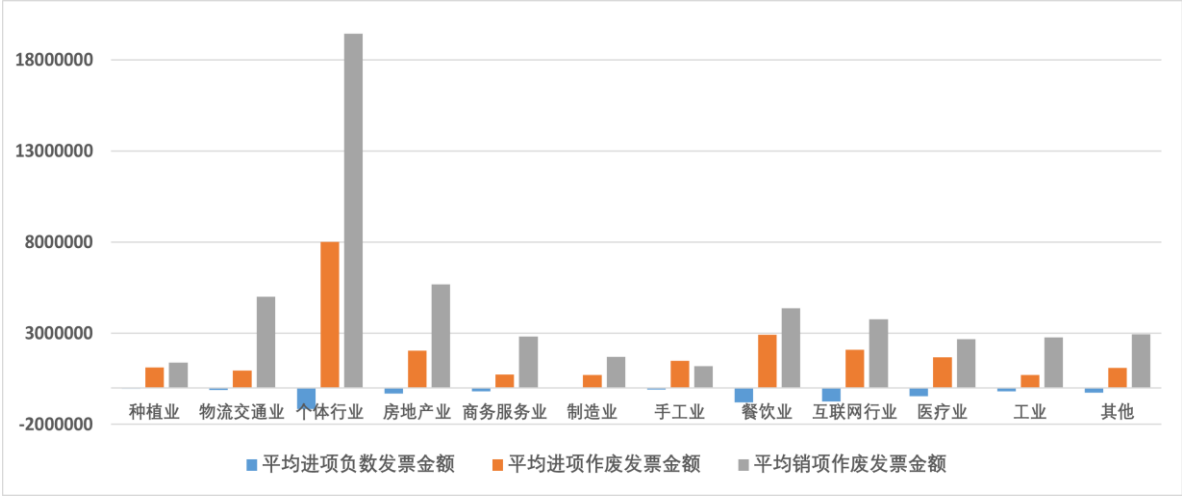


图 12 行业抗风险指标数据图

从图 1 中可以清晰看出，个体行业的平均销项作废金额要远高于其他行业，可能是由于个体行业人为操作失误较多，或是产品质量良莠不齐所导致的。而制造行业的作废发票金额最少，说明了行业发展较为稳定，行业整体管理能力强。接下来本文将利用 Python 使用模糊综合评价法(代码详见附录 9)对行业的抗风险能力进行定量的分析。

5.3.2 基于模糊综合评价的行业抗风险能力评估

(1) 模糊综合评价概述

模糊综合评价法是一种基于模糊数学的综合评价方法。模糊数学开创于控制论专家 LA.Zadeh 教授。可解决难以用复杂的、难以用数字关系描述的问题。

(2) 算法步骤

Step1：确定因素集 U 。

突发因素对于行业的影响(行业的抗风险能力)，可从多个方面进行综合评判，因此从数据发掘了行业中的平均进项负数发票金额 u_1 、平均进项作废发票金额 u_2 、平均销项作废发票金额 u_3 、平均利润 u_4 以及月利润方差 u_5 。

$$U = \{u_1, u_2, u_3, u_4, u_5\} \quad (22)$$

Step2：确定评语集 V 。

根据最终的隶属度，可确定抗风险能力的等级，因此确立 3 个评价等级，分别为 v_1 为强， v_2 为中， v_3 为弱。

$$V = \{v_1, v_2, v_3\} \quad (23)$$

Step3：确定模糊综合判断矩阵 R 。首先分析因素 u_i 的类型(如下表)，

表九 各个因素的类型

| 因素名称 | u_1 | u_2 | u_3 | u_4 | u_5 |
|------|-------|-------|-------|-------|-------|
| 因素类型 | 极大型 | 极小型 | 极小型 | 极大型 | 中间型 |

再根据因素类型，我们在模糊分布当中选择梯形分布，得到了每一个因素的隶属函数 F (如下图)。

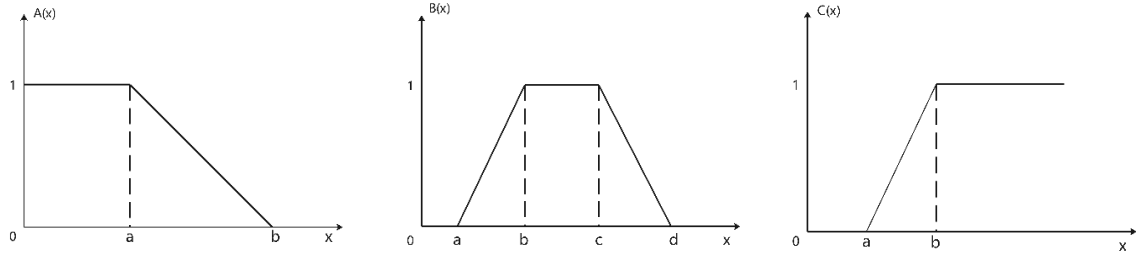


图 13 隶属函数

分别为极小型(图左)、中间型(图中)、极大型(图右)。进而将因素 u_{ij} 映射到隶属函数中，根据模糊变换得到各个因素的模糊综合判断矩阵。判断矩阵 R 的计算如下式：

$$R = F(U) \rightarrow F(V)$$

$$R = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \\ R_5 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1 \ 12} \\ r_{21} & r_{22} & \cdots & r_{2 \ 12} \\ \vdots & \vdots & \ddots & \vdots \\ r_{51} & r_{52} & \cdots & r_{5 \ 12} \end{bmatrix} \quad (24)$$

Step4: 熵权法确定权重集 A 。

得到各个因素的权重如下式：

$$\begin{aligned} A &= \{a_1, a_2, a_3, a_4, a_5\} \\ &= \{0.0505, 0.2971, 0.2491, 0.1282, 0.2751\} \end{aligned} \quad (25)$$

Step5: 综合判断。

最后根据模糊综合判断矩阵以及权重矩阵得到最终的综合评判结果 B 。

$$B = A * R \quad (26)$$

5.3.3 模型的求解

将预处理好的数据带入建立的抗风险能力评估模型中，得到每个行业的抗风险能力(数值越大抗风险能力越强)，结果如下图所示：

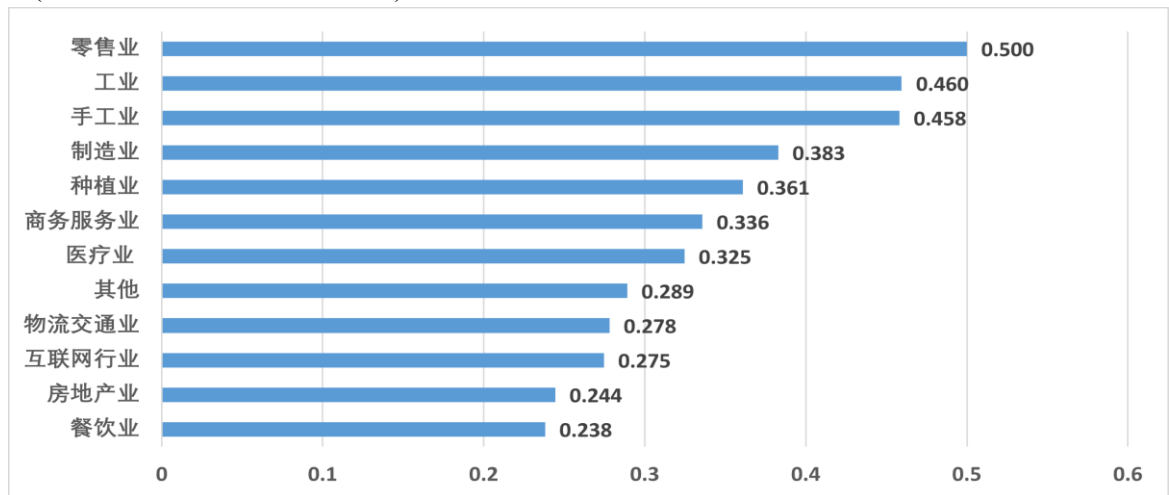


图 14 行业得分图

分析图 14 中的信息，零售业的抗风险能力最强，可能是由于零售业的流水较小，收入比较稳定导致的，工业和手工业次之。而餐饮行业的抗风险能力最弱。

将得到的抗风险能力数值作为新的评价指标，代入之前建立的信贷风险评估模型中，得到的指标权重结果如下所示：

表十 指标权重表

| 评价指标 | 企业净利润 | 交税金额 | 月度净利润方差 | 信誉评级 | 抗风险能力 |
|------|--------|--------|---------|--------|--------|
| 权重 | 0.0545 | 0.6079 | 0.0050 | 0.0452 | 0.2871 |

在加入抗风险能力的评价指标后，交税金额和抗风险能力两项之和占据了权重的 88%，说明在收到未知因素的影响下企业的信贷风险的大小主要收到以往交税的金额和抗风险能力的影响。

在得到各个企业的信贷风险得分后，依照问题一的思想，截取前 94%给予贷款资格，并按得分所占前 94%总得分的比例分配贷款比例，排名前十的结果如下表所示(详细结果见附录 8)：

表十一 企业信贷风险得分

| 排名 | 企业代号 | 风险得分 | 信誉评级 | 贷款比例 |
|----|------|--------|------|--------|
| 1 | E125 | 0.0109 | C | 0.0100 |
| 2 | E124 | 0.0105 | B | 0.0100 |
| 3 | E129 | 0.0098 | A | 0.0100 |
| 4 | E131 | 0.0080 | A | 0.0082 |
| 5 | E127 | 0.0073 | A | 0.0075 |
| 6 | E126 | 0.0072 | A | 0.0073 |
| 7 | E193 | 0.0070 | A | 0.0071 |
| 8 | E132 | 0.0069 | A | 0.0071 |
| 9 | E138 | 0.0067 | A | 0.0069 |
| 10 | E125 | 0.0067 | A | 0.0069 |

对表十一中的数据进行分析，排名前 10 的信誉评级为 A 的占了 8 个，说明信誉评级对企业的信贷风险的影响较大。

由于贷款额度最高为 100 万元，因此将超出金额的企业贷款比例修改为 0.01。将企业的贷款金额按照信誉评级相加，使用网格搜索法计算银行利益最大时，三种评级的利率情况，结果如下表所示：

表十二 银行决策表

| 信誉评级 | 贷款期限 | 对应利率 | 银行收益(万元) |
|------|------|--------|----------|
| A | 1 年 | 0.0465 | 407.32 |
| B | 1 年 | 0.0585 | |
| C | 1 年 | 0.0585 | |

由上表可见，在银行年贷款额限定的情况下，银行对信誉为 A 的企业给予 0.0465 的利率，对信誉为 B 的企业给予 0.0585 的利率，对信誉为 C 的企业给予 0.0585 的利率，此时银行的收益最大，最大值为 0.04073 亿元。

六、模型的评价、改进与推广

6.1 模型的优点

- 1, 熵权法确定指标的权重, 不存在主观因素, 使得 topsis 法建立评价模型更具客观性。
- 2, 该 bp 神经网络使用了 5 个指标, 在多次试验后, 测试正确率在 74% 上下波动, 使预测模型更具真实性。
- 3, 考虑到企业自身对未知因素的抗风险能力, 模型更具合理性。

6.2 模型的缺点

- 1, bp 神经网络训练中, 由于题目所给数据指标较少, 并且之间可能有所联系, 导致预测的结果没有达到最理想结果。
- 2, 未知因素对企业的影响无法量化分析, 导致模型分析结果可能与真实情况有所误差。

6.3 模型的推广

可推广到大型企业的风险投资风险评估、投资人对企业的投资风险评估。

七、参考文献

- [1]崔建荣.青海省“十五”期间中小企业信贷政策实施情况调查[J].青海金融,2006(05):30-32.
- [2]马刚宁.企业高质量发展的指标分析评价体系构建研究[J].商业会计,2021(14):35-38.
- [3]张景阳,潘光友.多元线性回归与 BP 神经网络预测模型对比与运用研究[J].昆明理工大学学报(自然科学版),2013,38(06):61-67.
- [4]周晓丹.外向型中小企业抗风险能力评价研究[J].商业文化(上半月),2011(03):151-152.

附录

附录 1

介绍：数据预处理结果

| 得分情况 | 销项总金额 | 进项总金额 | 净利润 | 交税金额 | 方差 | |
|----------|-------------|-------------|--------------|----------|-------------|-----|
| 0.04092 | 4065843302 | 6637942028 | -2572098727 | 6.33E+08 | 5.86065E+15 | E2 |
| 0.018761 | 590841712.8 | 162487956.6 | 428353756.2 | 35453921 | 1.00958E+14 | E3 |
| 0.017118 | 570177995.3 | 54179352.52 | 515998642.8 | 90892283 | 1.16103E+14 | E4 |
| 0.016281 | 1839969690 | 254584214.6 | 1585385475 | 3.07E+08 | 4.73536E+15 | E5 |
| 0.015971 | 202632288.9 | 227331979 | -24699690.1 | 30658411 | 3.94502E+12 | E6 |
| 0.01588 | 366439288.3 | 321178563.5 | 45260724.8 | 29458150 | 3.41348E+13 | E7 |
| 0.015835 | 512950626.5 | 63491942.06 | 449458684.4 | 83280070 | 9.14286E+13 | E8 |
| 0.015818 | 358888832.8 | 170418693.9 | 188470138.9 | 21905055 | 1.98565E+13 | E9 |
| 0.015696 | 319592135.4 | 26333514.09 | 293258621.3 | 49819903 | 7.10187E+12 | E10 |
| 0.015541 | 340831584.5 | 5417961.64 | 335413622.8 | 10629947 | 2.90772E+14 | E11 |
| 0.015528 | 148439127.1 | 144211948.1 | 4227179.06 | 14107918 | 1.08704E+13 | E12 |
| 0.015491 | 223684403.1 | 95447012.7 | 128237390.4 | 19318617 | 7.29687E+13 | E13 |
| 0.015474 | 212454771 | 103628503 | 108826268 | 31523252 | 2.79648E+13 | E14 |
| 0.015442 | 209084031.8 | 125103670.2 | 83980361.55 | 32676566 | 8.96474E+12 | E15 |
| 0.015429 | 211824339.2 | 4803944.44 | 207020394.8 | 7776513 | 3.82689E+13 | E16 |
| 0.015424 | 209410435.5 | 292748.42 | 209117687.1 | 4146250 | 2.81818E+13 | E17 |
| 0.015402 | 154980771.4 | 142782019.2 | 12198752.25 | 13107286 | 5.09651E+12 | E18 |
| 0.015382 | 187726474.5 | 137752837.2 | 49973637.33 | 9929916 | 1.91169E+13 | E19 |
| 0.015382 | 180443886.1 | 212140557.3 | -31696671.23 | 28671283 | 3.33719E+12 | E20 |
| 0.015376 | 191826403.6 | 204339817.2 | -12513413.57 | 30295655 | 6.78033E+12 | E21 |
| 0.015371 | 118412938.2 | 123194760.1 | -4781821.85 | 11309908 | 8.18074E+12 | E22 |
| 0.015319 | 111933993.5 | 74890175.94 | 37043817.58 | 10952470 | 7.43953E+12 | E23 |
| 0.015316 | 183238932.1 | 200994746.1 | -17755814.07 | 28805660 | 2.78669E+12 | E24 |
| 0.015313 | 140520367.7 | 86230669.41 | 54289698.25 | 13728085 | 3.79633E+12 | E25 |
| 0.015312 | 37997036.42 | 10004188.87 | 27992847.55 | 5836760 | 1.45414E+12 | E26 |
| 0.015311 | 36648003.49 | 42645410.52 | -5997407.03 | 5781067 | 1.63934E+11 | E27 |
| 0.015309 | 44058220.5 | 48889674.19 | -4831453.69 | 4601547 | 3.41627E+11 | E28 |
| 0.015308 | 49029625.43 | 1414753.47 | 47614871.96 | 4743903 | 3.26811E+12 | E29 |
| 0.010421 | 45171435.47 | 186618.44 | 44984817.03 | 1804943 | 5.02063E+12 | E30 |
| 0.010414 | 71493447.69 | 24220896.36 | 47272551.33 | 6615074 | 5.7363E+12 | E31 |
| 0.010386 | 47282037.6 | 5059207.58 | 42222830.02 | 6989811 | 2.70216E+11 | E32 |
| 0.010221 | 44150482.29 | 2542575.72 | 41607906.57 | 1324771 | 1.81409E+12 | E33 |
| 0.010093 | 69611432.64 | 69644756.16 | -33323.52 | 4828964 | 2.78112E+12 | E34 |

附录 2

介绍：改代码使用 matlab 编写，用于计算问题一、二、三中 Topsis 得分

```

% clear;clc
%% 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]:  '); % [2,1,3]
    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 % 如果之前标准化后的 Z 矩阵中存在负数，则重新对 X 进行标准化
            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 进行标准化')
            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
            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: ')
    end
end

```

```

        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，则你需要输入[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;
        while OK == 0
            if abs(sum(weight) -1)<0.000001 && size(weight,1) == 1 &&
size(weight,2) == m % 注意，Matlab 中浮点数的比较要小心
                OK =1;
            else
                weight = input('你输入的有误，请重新输入权重行向量: ');
            end
        end
    end
end
else
    weight = ones(1,m) ./ 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
disp('最后的得分为: ')
stand_S = S / sum(S)
[sorted_S,index] = sort(stand_S,'descend')

```

附录 3

介绍 python 网络搜索法代码

```

for akey,avalue in A.items():
    for bkey,bvalue in B.items():
        for ckey,cvalue in C.items():
            sum=0.317964*akey*(1-avalue)+0.379757*bkey*(1-
bvalue)+0.300198*ckey*(1-cvalue)
            if(sum>max):
                max=sum
                a=akey
                b=bkey
                c=ckey

```

| 附录 4 | |
|--------------------------------------|----------|
| 介绍：本表格是附件 2 中 302 家企业的信誉预估等级，具体见支撑材料 | |
| 企业代 号 | 信誉等 级 |
| E124 | 73.28669 |
| E125 | 53.63663 |
| E126 | 107.4653 |
| E127 | 87.81984 |
| E128 | 91.27201 |
| E129 | 95.88482 |
| E130 | 78.32368 |
| E131 | 90.35146 |
| E132 | 92.38489 |
| E133 | 78.09453 |
| E134 | 85.91625 |
| E135 | 86.06216 |
| E136 | 77.63985 |
| E137 | 80.02616 |
| E138 | 83.96389 |
| E139 | 95.35519 |
| E140 | 92.71599 |
| E141 | 88.10112 |
| E142 | 78.99785 |
| E143 | 79.45387 |
| E144 | 81.77133 |
| E145 | 77.63006 |
| | |
| E146 | 76.70745 |
| E417 | 66.13369 |
| E418 | 66.1894 |
| E419 | 65.12539 |
| E420 | 65.95595 |
| E421 | 66.15998 |
| E422 | 66.12514 |
| E423 | 66.12727 |
| E424 | 66.11094 |
| E425 | 66.12997 |

| 附录 5 |
|--|
| 介绍：使用 matlab 编写的 BP 神经网络代码，预测 302 家企业的信誉等级（具体见支撑材料） |
| <pre>function [Y,Xf,Af] = bpnetwork(X,~,~) % Input 1 x1_step1.xoffset = [196919.86;280;-2572098727;0;90226588.48];</pre> |

```

x1_step1.gain = [4.91926697999563e-10;3.01298215008078e-10;4.81060156293048e-
10;3.16060551100358e-09;3.41259080612328e-16];
x1_step1.ymin = -1
% Layer 1
b1 = [-3.6716762033191354675;-3.4551594387607105219;.....]
IW1_1 = [1.7699702073553171822 -1.6850186791776964323 ...]

% Layer 2
b2 = 0.60021767942045056721;
LW2_1 = [0.23011421795538128388 0.0970 ...]

% Output 1
y1_step1.ymin = -1;
y1_step1.gain = 0.1;
y1_step1.xoffset = 10;

% ===== SIMULATION =====

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX
    X = {X};
end

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},1); % samples/series
else
    Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    X{1,ts} = X{1,ts}';
    Xp1 = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

```



```

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1
Y{1,ts} = mapminmax_reverse(a2,y1_step1);
Y{1,ts} = Y{1,ts}';
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX
    Y = cell2mat(Y);
end
end

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end

```

附录 6

介绍 python 行业分类代码

```

for key,value in hangye.items():
    if("园"in value or "桃" in value or "花" in value or "农" in value):
        plant.append(key)

```

```

        continue
    if("物流"in value or "交通" in value or "运" in value ):
        tradstation.append(key)
        continue
    if("个体"in value):
        tail.append(key)
        continue
    if("房" in value or "建" in value or "工程" in value):
        build.append(key)
        continue
    if("贸"in value or "服务"in value or "劳务" in value or "装" in value or "咨询"in
value or "律师" in value or "管理"in value or "营销"in value or "维修"in value or "代理
"in value):
        serve.append(key)
        continue
    if("制造" in value or "机械" in value or "金"in value or "材料"in value or "机"in
value or "设备"in value):
        create.append(key)
        continue
    if("厂" in value or "衣" in value or"鞋" in value or"纸"in value or"家"in value or "材
"in value or "塑胶"in value or "轮"in value or "织"in value or "地毯"in value or"电器"in
value or "办公"in value):
        process.append(key)
        continue
    if("食" in value or "餐" in value or "生"in value or "蟹" in value):
        eat.append(key)
        continue
    if("电子"in value or "科" in value or "网络"in value or "传"in value or "信息" in
value or "技"in value):
        internet.append(key)
        continue
    if("药"in value or "医" in value):
        medicine.append(key)
        continue
    if("气"in value or "油" in value or "石"in value):
        industion.append(key)
        continue
    other.append(key)

```

附录 7

介绍 python 行业分类结果

| | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|
| E179 | E142 | E124 | E134 | E144 | E136 | E196 | E141 | E143 | E195 | E180 | E167 |
| E267 | E152 | E125 | E135 | E145 | E182 | E224 | E175 | E162 | E251 | E247 | E171 |

| | | | | | | | | | | | |
|------|------|------|------|------|------|------|------|------|------|------|------|
| E329 | E163 | E126 | E137 | E148 | E188 | E250 | E181 | E165 | E261 | E299 | E228 |
| E336 | E172 | E127 | E140 | E154 | E197 | E287 | E185 | E176 | E379 | | E232 |
| E387 | E201 | E128 | E146 | E173 | E266 | E312 | E397 | E194 | | | E252 |
| E417 | E258 | E129 | E147 | E178 | E277 | E315 | | E204 | | | E256 |
| | E289 | E130 | E149 | E190 | E283 | E325 | | E213 | | | E260 |
| | E292 | E131 | E150 | E191 | E290 | E339 | | E220 | | | E282 |
| | E313 | E132 | E151 | E198 | E295 | E341 | | E222 | | | E286 |
| | E321 | E133 | E157 | E219 | E302 | E355 | | E223 | | | E293 |
| | E340 | E138 | E158 | E221 | E334 | E357 | | E227 | | | E303 |
| | E414 | E139 | E160 | E225 | E335 | E375 | | E243 | | | E331 |
| | | E153 | E161 | E229 | E342 | E383 | | E248 | | | E353 |
| | | E155 | E166 | E233 | E343 | E400 | | E259 | | | E354 |
| | | E156 | E168 | E234 | E344 | E409 | | E271 | | | E394 |
| | | E159 | E169 | E246 | E347 | E413 | | E284 | | | E408 |
| | | E164 | E170 | E257 | E350 | | | E304 | | | |
| | | E187 | E174 | E269 | E358 | | | E317 | | | |
| | | E193 | E177 | E278 | E391 | | | E328 | | | |
| | | E200 | E183 | E279 | E403 | | | E345 | | | |
| | | E202 | E184 | E281 | E411 | | | E351 | | | |
| | | E205 | E186 | E298 | E421 | | | E364 | | | |
| | | E207 | E189 | E301 | E423 | | | E365 | | | |
| | | E208 | E192 | E305 | | | | E367 | | | |
| | | E211 | E199 | E308 | | | | E368 | | | |
| | | E217 | E203 | E310 | | | | E369 | | | |
| | | E235 | E206 | E311 | | | | E371 | | | |
| | | E236 | E209 | E318 | | | | E377 | | | |
| | | E237 | E210 | E324 | | | | E392 | | | |
| | | E238 | E212 | E326 | | | | E395 | | | |
| | | E239 | E214 | E330 | | | | E396 | | | |
| | | E240 | E215 | E332 | | | | E401 | | | |
| | | E241 | E216 | E338 | | | | E410 | | | |
| | | E242 | E218 | E348 | | | | E416 | | | |
| | | E244 | E226 | E349 | | | | E419 | | | |
| | | E255 | E230 | E370 | | | | | | | |
| | | E262 | E231 | E372 | | | | | | | |
| | | E264 | E245 | E378 | | | | | | | |
| | | E270 | E249 | E382 | | | | | | | |
| | | E272 | E253 | E388 | | | | | | | |
| | | E273 | E254 | E389 | | | | | | | |
| | | E280 | E263 | E393 | | | | | | | |
| | | E285 | E265 | E398 | | | | | | | |
| | | E288 | E268 | E402 | | | | | | | |
| | | E294 | E274 | E405 | | | | | | | |

| 附录 8 | | | | | | | | |
|--------------|-----------------|-----------------|------------------|----------------|--------------|----------------|-----------------|--------------|
| 介绍 问题三风险得分结果 | | | | | | | | |
| 企业 代号 | 销项总 金额 | 进项总 金额 | 净利润 | 交税金 额 | 方差 | 信誉等 级 | 抗风险 因素 | 得分 |
| E191 | 2561384 36.3 | 3041036 52.2 | - 4.8E+0 7 | 423945 45.2 | 2.19E +14 | 61.5165 419 | 0.335695 348 | 0.013 268 |
| E127 | 6519379 54.5 | 1765558. 94 | 6.5E+0 8 | 195702 21 | 2.12E +14 | 87.8198 368 | 0.500000 009 | 0.011 375 |
| E125 | 9412113 64.5 | 1001201 809 | - 6E+07 | 836297 45.5 | 7.9E+ 13 | 53.6366 286 | 0.500000 009 | 0.009 288 |
| E124 | 7417803 67.8 | 8442027 97.2 | - 1E+08 | 666573 80.3 | 7.27E +13 | 73.2866 871 | 0.500000 009 | 0.008 273 |
| E128 | 2423901 51.8 | 9360499. 95 | 2.33E+ 08 | 829906 7 | 9.24E +13 | 91.2720 148 | 0.500000 009 | 0.006 127 |
| E129 | 3248060 32.6 | 8557780 6.82 | 2.39E+ 08 | 495930 39.7 | 2.56E +13 | 95.8848 247 | 0.500000 009 | 0.005 864 |
| E126 | 5206910 97.8 | 1288187 23.6 | 3.92E+ 08 | 173162 78.6 | 6.93E +13 | 107.465 318 | 0.500000 009 | 0.005 69 |
| E160 | 2228334 68.6 | 1698956 37 | 52937 832 | 262432 01.2 | 2.01E +13 | 88.0066 683 | 0.244331 141 | 0.005 639 |
| E153 | 2410237 67.8 | 93077 | 2.41E+ 08 | 136433 34.5 | 7.44E +13 | 93.7565 613 | 0.500000 009 | 0.005 45 |
| E162 | 1369611 59.2 | 1433228 78.7 | - 63617 19 | 142909 20.1 | 4.22E +13 | 74.7258 731 | 0.274549 42 | 0.005 27 |
| E177 | 9771736 0.95 | 7940970 5.98 | 18307 655 | 727244 0.28 | 2.89E +13 | 73.8381 069 | 0.244331 141 | 0.004 979 |
| E147 | 1876697 83.1 | 1564037 12.1 | 31266 071 | 121473 20.3 | 1.66E +13 | 83.3940 195 | 0.244331 141 | 0.004 924 |
| E161 | 1539000 98.9 | 963190.5 6 | 1.53E+ 08 | 623179 3.01 | 1.92E +13 | 88.9383 61 | 0.244331 141 | 0.004 917 |
| E137 | 1432643 72.9 | 1083014 41.8 | 34962 931 | 102737 00.2 | 1.97E +13 | 80.0261 646 | 0.244331 141 | 0.004 9 |
| E140 | 1824290 33.1 | 7398424. 69 | 1.75E+ 08 | 576463 2.93 | 1.56E +13 | 92.7159 924 | 0.244331 141 | 0.004 885 |
| E146 | 1311106 08.4 | 1373281 70.1 | - 62175 62 | 120059 24 | 1.64E +13 | 76.7074 476 | 0.244331 141 | 0.004 845 |
| E151 | 1024211 53 | 1185514 2.48 | 90566 010 | 640998 6.19 | 1.6E+ 13 | 81.0844 146 | 0.244331 141 | 0.004 75 |

| | | | | | | | | |
|------|-----------------|-----------------|--------------|----------------|--------------|----------------|-----------------|--------------|
| E135 | 1355375 58.5 | 2638095 7.44 | 1.09E+ 08 | 775832 1.61 | 7.04E +12 | 86.0621 604 | 0.244331 141 | 0.004 685 |
| E149 | 1012406 06.3 | 637449.9 4 | 1.01E+ 08 | 591734 0.4 | 1.21E +13 | 82.1182 038 | 0.244331 141 | 0.004 68 |
| E174 | 9115003 6.85 | 8687494 0.41 | 42750 96 | 140856 70.1 | 1.71E +12 | 76.6636 041 | 0.244331 141 | 0.004 679 |
| E199 | 5957554 2.09 | 169812.2 3 | 59405 730 | 259226 5.62 | 2E+13 | 74.1749 905 | 0.244331 141 | 0.004 678 |
| E134 | 1228154 70 | 4295229. 39 | 1.19E+ 08 | 736893 0.27 | 6.14E +12 | 85.9162 457 | 0.244331 141 | 0.004 665 |
| E185 | 9632902 1.71 | 788684.1 4 | 95540 338 | 492024 6.05 | 6.72E +12 | 81.7757 962 | 0.238063 433 | 0.004 643 |
| E166 | 7761867 4.3 | 3190440. 61 | 74428 234 | 382540 7.09 | 1.41E +13 | 77.6849 98 | 0.244331 141 | 0.004 62 |
| E175 | 7400182 3.99 | 8911908 7.08 | 1.5E+0 7 | 111388 18.5 | 2.08E +11 | 73.4371 503 | 0.238063 433 | 0.004 614 |

附录 9

介绍 python 行业分类结果

```
# utf-8
# 12 个行业
# 种植业 物流交通业 零售业 房地产业 商务服务业 制造业 手工业
# 餐饮业 互联网行业 医疗业 工业 其他
# tree, traffic, retail, room, service, manu, handwork, food,
it, medical, industry, other
# 0 1 2 3 4 5 6
7 8 9 10 11
```

梯形极小型

def trapzoid_Small(X,a,b): # 处理一维数据 输出一维矩阵

```
Y = []
for x in X:
    if x < a:
        y = 1
        Y.append(y)
    elif a <= x <= b:
        y = (b - x) / (b - a)
        Y.append(y)
    elif x > b:
        y = 0
        Y.append(y)
return Y
```

```

# 梯形中间型
def trapezoid_Intermedius(X,a,b,c,d):
    Y = []
    for x in X:
        if x < a:
            y = 0
            Y.append(y)
        elif a<= x < b:
            y = (x -a)/(b-a)
            Y.append(y)
        elif b<= x < c:
            y = 1
            Y.append(y)
        elif c<= x < d:
            y = (d-x)/(d-c)
            Y.append(y)
        elif x >= d:
            y = 0
            Y.append(y)
    return Y

# 梯形极大型
def trapezoid_Larger(X,a,b):
    Y = []
    for x in X:
        if x < a:
            y = 0
            Y.append(y)
        elif a<= x <=b:
            y = (x-a)/(b-a)
            Y.append(y)
        elif x > b:
            y = 1
            Y.append(y)
    return Y

# fuzzy comprehensive evaluation method
def FCEM():
    # 1. 确定因素集  $U = \{u_1, u_2, u_3, u_4, u_5, \dots\}$ 
    #  $u_1$ -平均进项负数发票金额-该行业购买订单进行一半取消  $u_2$ -平均进项
    # 作废发票金额-行业购买订单取消
    #  $u_3$ -平均销项作废发票金额-行业卖出的订单被取消  $u_4$ -平均利润-行业
    # 平均利润  $u_5$ =平均方差-行业的月利润方差
    u1 = [-43833.81667,-123008.9517,-1160043.928,-309592.3097,-188064.8715,-
16009.76087,-103119.1338,-790842.478,-739973.9723,-467547.37,-203013.3033,-

```

```

271174.6544]
    u2
    =
[1120738.457,957724.855,7994954.831,2051952.331,735620.7206,695683.4596,148751
0.872 ,2914387.058, 2098569.179 ,1675371.555 ,704016.6233
,1104264.459]
    u3 = [1389503.182 ,5009995.624 ,19422827.94, 5667835.324
,2825051.763 ,1694995.253 ,1191328.014 ,4374174.772 ,3764767.735
,2670662.558 ,2776311.09,2938046.781]

    u4 = [8919361.762, 26350291.62 ,49614527.89, 24600597.94
,5167411.92,4236546.546, -3292916.13,35619231.48, 9235189.574,
8151556.508 ,1004774.677 ,7216245.727]

    u5
    =
[635727786813.874 ,1104897573813.08 ,14890067978879.8 ,4462374565192.52 ,57611
21368117.69 ,559165546433.276,
367352658670.718,1909590767452.6,2006347569721.74,517381398543,137568631582
,2726495094813.47]
    # U = [u1,u2,u3,u4,u5]

    # 2. 确定评语集 V = {v1,v2,v3}
    # 对突发因素的抗风险能力 v1 = "强" v2 = "中" v3 = "弱"
    v1 , v2 , v3 = "强" , "中" , "弱"
    # V = [v1,v2,v3]

    # 3. 确定模糊综合判断矩阵 R ----确定每个 u 对每个 v 的关系, 及确定隶属函
    数 得到隶属度-[0~1] 方法: 1.模糊统计法 2.借助已有的客观尺度 3.指派法
    # 指派法: 1.确定因素类型 u2,u3-极小型 u1,u4-极大型 u5-中间型 2.
    在模糊分布中选择分布, 得到函数(符合正向化)
    # 梯形分布
    R1 = trapezoid_Small(u1,-1160044,-43834) # a:20%为 1 80%
    在中间 b=最小值
    R2 = trapezoid_Small(u2,704017,7994955) # a:20%为 1 80%
    在中间 b=最大值
    R3 = trapezoid_Small(u3,1389504,19422828) # a:20%为 1 80%
    在中间 b=最大值

    R4 = trapezoid_Larger(u4,-3292916.13, 35619230) # a:最小值 b=最
    大的
    R5 = trapezoid_Intermedius(u5,137568631582,1104897573812,
    1909590767453,14890067978880) #a=最小值 b=中位数 c=中位数 d=最大值
    R = [R1,R2,R3,R4,R5]
    # 隶属函数绘图
    # import matplotlib.pyplot as plt

```

```

# plt.plot(u2,R2,label="u1")
# plt.xlabel("x")
# plt.ylabel("y")
# plt.legend()    #打上标签
# plt.show()
# Ri 如果加起来大于 1 需要归一化
for i in range(len(R[0])):
    s = []
    for j in range(len(R)):
        s.append(R[j][i])
    if sum(s) != 1:
        for k in range(len(R)):
            R[k][i] = R[k][i] / sum(s)

# 4. 确定各个 U 的权重 A = {a1,a2,a3,a4,a5.....} 层次分析法 熵权法 德菲尔法
等都可以
# Matlab 熵权法
A = [0.0505272795328914, 0.297148658170007, 0.249097452045129,
0.128166963628809, 0.275059646623163]
#          u1                      u2                      u3
u4                      u5

# 5.综合判断 , 得到模糊向量 B  B = A*R
B1 = [ A[0]*r for r in R[0]]
B2 = [ A[1]*r for r in R[1]]
B3 = [ A[2]*r for r in R[2]]
B4 = [ A[3]*r for r in R[3]]
B5 = [ A[4]*r for r in R[4]]
B = [B1,B2,B3,B4,B5]
MAX_B = []
for i in range(len(B[0])):
    s = []
    for j in range(len(R)):
        s.append(R[j][i])
    MAX_B.append(max(s))
print(MAX_B)

return B
ans = FCEM()
print(ans)

```