

# “华为杯”第十五届中国研究生

## 数学建模竞赛

### 题 目                      对恐怖主义袭击事件数据的量化分析

---

#### 摘                      要：

恐怖主义是人类共同威胁，打击恐怖主义是每个国家应该承担的责任。对恐怖袭击事件相关数据的深入分析有助于加深人们对恐怖主义的认识，为反恐防恐提供有价值的信息支持。

**依据危害性对恐怖袭击事件分级(问题 1)** 首先对数据库原始数据进行预处理，包括无关信息剔除，缺失信息填充，数据归一化，获得 14 维可能与恐怖袭击事件分级相关的特征向量。利用 PCA 算法对特征向量进行进一步地特征提取和降维最终获得 10 维与恐怖袭击事件分级关联性较强的特征向量。在已知分级为 5 类的前提下，提出了一种基于初始质心选取优化的 K—means 改进算法，经检得其 DVI 系数为 0.04262，轮廓系数为 0.3950，改进后 K—means 算法相比于传统方法就有更好的分类准确性。基于改进后的聚类算法，可得各典型事件危害级别如表 4.7 所示。选取恐怖袭击事件分级最高的集合（1 级），再次利用 PCA 算法对于各恐怖袭击事件进行危害等级打分排序，建立量化的评价系统，最终获得近二十年来危害程度最高的十大恐怖袭击事件如表 4.11 所示。基于聚类结果的 PCA 恐怖袭击排序有利于降低运算量，提高排名准确率。

**依据事件特征发现恐怖袭击事件制造者(问题 2)** 根据问题要求界定处理对象的范围；分析数据源信息，剔除冗余的数据信息，对数据进行降维处理，并筛选出核心关联要素特征，并将其数值规范化处理；采用逻辑回归的数据挖掘方法，对事件样本进行数据关联，进而建立逻辑回归预测模型，计算得出恐怖袭击事件的分类，根据危害性大小排序标出前 5 号组织，并对问题给出的恐怖袭击事件进行了嫌疑人的分析排序；最后对逻辑回归预测模型的结果进行 ROC 曲线分析，验证了本文所建立逻辑回归预测模型的准确度。

**对未来反恐态势的分析(问题 3)** 为了提高数据处理效率，同时兼顾准确性的要求，以三个月（一个季度）为统计步长对 2015 年—2017 年的各指标进行提

取，并对三年来的变化规律进行可视化统计分析。考虑到新提取的统计数据，带有明显的时间序列特征，同时在数据规模属于小样本数据，故考虑采用灰色预测模型进行建模分析，并对未来各地区恐怖袭击事件进行预测。结果表明，采用灰色预测算法相比较于数据拟合方法，具有较高的预测准确性，其和方差为 0.2143，均方根为 0.158，确定系数为 0.9223。最后根据可视化的统计规律和灰色预测模型，给出了反恐斗争的建议和见解。

**数据的进一步利用（问题 4）** 根据问题 1 中改进的 K—means 算法获得了各类恐怖袭击事件的分级标签，原本无监督学习样本，转化成了有监督的学习样本，我们反过来对获得分级标签的恐怖袭击事件样本建立有监督的机器学习模型，利用 Matalab 机器学习分类工具箱设计不同的分类器（共 22 种）对数据进行训练，结果表明不同机器学习模型最高识别准确率为 89.4，测试结果最高识别准确率为 87.8%，反过来证明了问题 1 中聚类结果具有较高的准确性。同时采用 NCA 近邻分析法对各特征向量进行权重打分，其排序结果与问题 1 中 PCA 权重分析表现出很高的一致性，进一步验证了基于 PCA 的特征向量提取与分级权重计算的有效性。

**关键词：**改进 k-means 主成分分析 逻辑回归 灰色预测 机器学习

## 目录

一、前言 .....	5
1.1 研究背景.....	5
1.2 问题重述.....	5
二、模型假设 .....	6
三、模型符号分析与说明 .....	6
四、问题一的模型建立与求解 .....	7
4.1 问题分析.....	7
4.2 数据预处理.....	8
4.2.1 信息剔除与数据化 .....	8
4.2.2 缺失数据填充与处理 .....	10
4.2.3 数据标准化 .....	11
4.3 恐怖袭击分级特征向量提取.....	12
4.4 聚类分析-事件分级 .....	16
4.4.1 聚类分析方法调研 .....	17
4.4.2 改进的 k-means 算法.....	18
4.4.3 聚类结果求解 .....	18
4.5 主成分分析排序-前十恐怖袭击事件 .....	21
<b>4.6 模型检验 .....</b>	<b>23</b>
4.6.1 聚类结果评价 .....	23
4.6.2 相关性检验 .....	24
五、问题二的模型建立与求解 .....	24
5.1 问题分析.....	24
5.2 数据处理.....	25
5.2.1 范围界定筛选 .....	26
5.2.2 属性特征确定 .....	26
5.3.3 信息规范化 .....	27
5.3 恐怖主义事件关联规则挖掘.....	27

5.3.1 恐怖主义事件特征统计分析 .....	27
5.3.2 逻辑回归事件相似度计算 .....	29
5.3.3 逻辑回归预测模型确立 .....	30
5.3.4 利用逻辑回归预测模型进行求解 .....	30
5.4 模型评价.....	32
六、问题三的模型建立与求解 .....	33
6.1 问题分析.....	33
6.2 特征可视化统计分析.....	33
6.2.1 主要原因分析 .....	34
6.2.2 时空与蔓延特性分析 .....	36
6.3 基于灰色预测的恐怖袭击预测.....	38
6.4 模型验证.....	39
6.5 反恐斗争的见解和建议.....	40
七、问题四的模型建立与求解 .....	41
7.1 问题分析.....	41
7.2 模型建立.....	41
7.3 网络训练与测试.....	42
八、参考文献 .....	48
附录 .....	49

# 一、前言

## 1.1 研究背景

恐怖袭击是指极端分子或组织人为制造的、针对但不仅限于平民及民用设施的、不符合国际道义的攻击行为，它不仅具有极大的杀伤性与破坏力，能直接造成巨大的人员伤亡和财产损失，而且还给人们带来巨大的心理压力，造成社会一定程度的动荡不安，妨碍正常的工作与生活秩序，进而极大地阻碍经济的发展。

恐怖主义是人类共同威胁，各国政府、媒体和公众对此有着广泛的关注，打击恐怖主义是每个国家应该承担的责任。通过恐怖袭击事件相关数据的深入分析有助于为反恐防恐提供有价值的信息支持。

## 1.2 问题重述

基于上述研究背景，本文需研究完成以下任务：

### 任务 1：依据危害性对恐怖袭击事件分级

依据某组织搜集整理的全球恐怖主义数据库（GTD）中 1998-2017 年世界上发生的恐怖袭击事件的记录（附件 1）以及其它有关信息，结合现代信息处理技术，借助数学建模方法建立基于数据分析的量化分级模型，将附件 1 给出的事件按危害程度从高到低分为一至五级，列出近二十年来危害程度最高的十大恐怖袭击事件，并对部分要求的事件进行分级，并给出列表中事件的分级。

典型事件危害级别

事件编号	危害级别
200108110012	
200511180002	
200901170021	
201402110015	
201405010071	
201411070002	
201412160041	
201508010015	
201705080012	

### 任务 2：依据事件特征发现恐怖袭击事件制造者

针对在 2015、2016 年度发生的、尚未有组织或个人宣称负责的恐怖袭击事件，运用数学建模方法寻找是否可以将可能是同一个恐怖组织或个人在不同时间、不同地点多次作案的若干案件串联起来统一组织侦查的可能性，将可能是同一个恐怖组织或个人在不同时间、不同地点多次作案的若干案件归为一类，对应的未知作案组织或个人标记不同的代号，并按该组织或个人的危害性从大到小选出其中的前 5 个，记为 1 号-5 号。再对表中列出的恐怖袭击事件，按嫌疑程度对 5 个嫌疑人排序，并将结果填入表中（表中样例的意思是：对事件编号为 XX 的事件，

3 号的嫌疑最大，其次是 4 号，最后是 5 号)，如果认为某嫌疑人关系不大，也可以保留空格。

恐怖分子关于典型事件的嫌疑度

	1 号嫌疑人	2 号嫌疑人	3 号嫌疑人	4 号嫌疑人	5 号嫌疑人
样例 XX	4	3	1	2	5
201701090031					
201702210037					
201703120023					
201705050009					
201705050010					
201707010028					
201707020006					
201708110018					
201711010006					
201712010003					

### 任务 3：对未来反恐态势的分析

依据附件 1 并结合因特网上的有关信息，建立适当的数学模型，研究近三年来恐怖袭击事件发生的主要原因、时空特性、蔓延特性、级别分布等规律，进而分析研判下一年全球或某些重点地区的反恐态势，用图/表给出你们的研究结果，提出你们对反恐斗争的见解和建议。

### 任务 4：数据的进一步利用

通过数学建模还可以发挥附件 1 数据的哪些作用？给出模型和方法。

## 二、 模型假设

1. 恐怖事件包含太多缺失属性值的数据行包含有用信息的可能性较少；
2. 恐怖事件数据变化趋势相似时，认为它们包含的信息也相似；
3. 依据物元思想，假设恐怖主义袭击事件是“物”，恐怖主义袭击事件危害性影响因素的指标是“特征”，指标的值是“量值”。
4. 不考虑 GTD 数据库外其它因素对恐怖袭击事件危害性的影响

## 三、 模型符号分析与说明

表 3.1 模型符号分析与说明

符号	符号说明
$R$	相关系数
$S_i(A, B)$	相似度
$v_i(A)$	A 事件特征属性值
$O(n)$	计算复杂度
$P(linked)$	事件关联概率
$\alpha$	常量
$\beta_i$	逻辑回归
$X_n$	事件属性相似度
$n$	网络输入层节点数
$l$	隐含层节点数
$m$	输出层节点数
$w_{ij}, w_{jk}$	连接权值
$b$	输出层阈值
$\eta$	学习效率
$\omega$	权重
$e$	网络预测误差

## 四、问题一的模型建立与求解

### 4.1 问题分析

根据题目要求,灾难性事件的分级主要是按照人员伤亡和经济损失进行主观等级划分。对于恐怖事件的等级划分,需要考虑时机、地域等更多因素,并且需要建立基于数据分析的量化分级模型,而不是主观分级。

题目已给的条件是 GTD 全球恐怖主义数据库关于近 20 年的恐怖主义事件数据。对数据进行初步的分析,发现给出的数据存在大量冗余信息和缺失数据等数据缺陷问题,调研相关文献<sup>[1]</sup>,了解数据库挖掘的一些常用方法,应用聚类分析和主成分分析等方法完成问题一要求的恐怖主义等级划分和找出十大恐怖主义事件,并将表格中给定的 9 个编号事件进行分级。

由于题目给出的样本数量庞大,如果直接应用主成分分析法对所有恐怖事件进行综合得分排序,虽然比较容易得到前十大恐怖袭击案件,但是对于恐怖事件分为五级并不能很好地量化分析解决,而聚类分析方法对于事件分级具有更大的优势,结果也更直观,因此本文确定先用改进的 k-means 聚类方法将恐怖袭击事

件分为 5 级，确定 9 个编号事件的等级，然后在 1 级恐怖事件样本中进行主成分分析排序得到前十大恐怖袭击事件，这样可以大大缩小主成分分析的样本，提高计算效率和精确度。确定任务一的总体思路如图 4.1 所示。

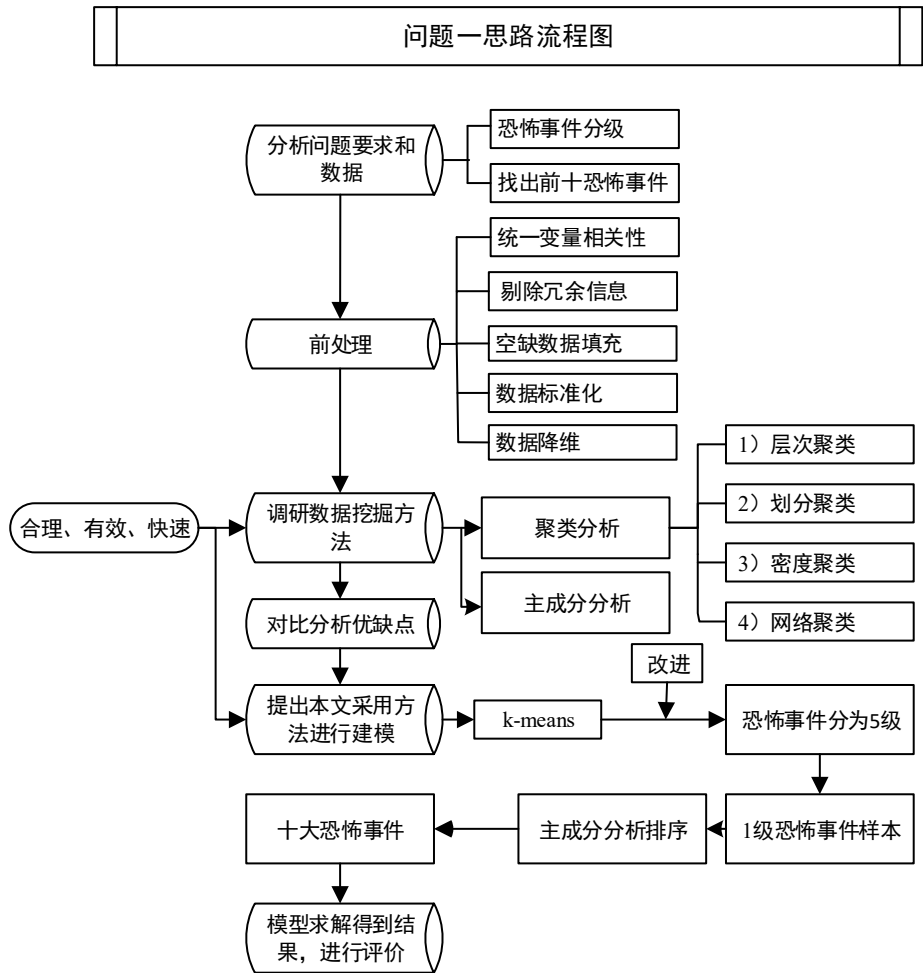


图 4.1 问题一思路流程图

## 4.2 数据预处理

### 4.2.1 信息剔除与数据化

对于附件 1 中所给的 Excel 进行分析，对附件 2 以及附件 1 中的解释说明掌握附件 1 中恐怖事件每个属性的含义。查阅相关文献，为了后续数据的处理，对附件 1 进行简单处理。保留对本任务数据分析处理有价值的数据信息，剔除明显没有数据价值意义的问题解释信息等栏，剔除部分栏如下表 4.1 所示，并将部分恐怖事件属性的文本描述转化为数字代码。

表 4.1 剔除信息列表

编号	剔除栏名称	含义
1	location	位置描述
2	summary	事件摘要



3	attacktype1_txt	攻击类型名称
4	attacktype2_txt	第二攻击类型名称
5	attacktype3_txt	第三攻击类型名称
6	targettype1_txt	目标/受害者类型名称
7	argsubtype1_txt	目标/受害者子类型名称
8	targettype2_txt	第二目标/受害者类型名称
9	targetsubtype2_txt	第二目标/受害者子类型名称
10	natlty2_txt	第二目标/受害者的国籍名称
11	argtype3_txt	第三目标/受害者类型名称
12	targetsubtype3_txt	第三目标/受害者子类型名称
13	natlty3_txt	第三目标/受害者的国籍名称
14	gsu bname	犯罪子集团名称
•	•	•
•	•	•
•	•	•
•	claimmode_txt	claimmode_txt
•	claimmode2_txt	声称负责的第二组织
•	claimmode2_txt	声称负责的第二组织的模式名称
•	scite1	第一引用来源
•	Scite2	第二引用来源
•	Scite3	第三引用来源
•	dbsource	数据收集

原始数据经过无用信息剔除和数据化处理之后，根据任务分析，结合给出的恐怖事件记录数据表，发现部分恐怖事件属性描述中存在着许多缺失的数据，以及部分恐怖事件的数据之间存在着极大的相似性，同时恐怖事件部分属性的数据量级不同。因此，该部对模型准备处理得到的表格数据进行预处理，数据预处理流程图如图 4.2 所示。

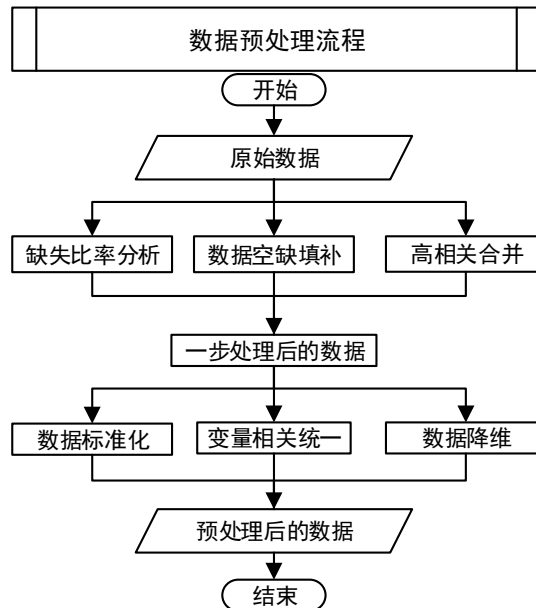


图 4.2 数据预处理流程图

#### 4.2.2 缺失数据填充与处理

##### (1) 缺失值比率分析

分析给出的恐怖事件数据，其中部分恐怖事件包含大量缺失属性数据，为得到高质量的数据挖掘效果，基于恐怖事件包含太多缺失属性值的数据行包含有用信息的可能性较少的假设。利用缺失值比率筛选，将恐怖事件数据行缺失属性值大于某个阈值的行去掉，以提高恐怖事件数据的质量<sup>[2]</sup>。例如凶手人数已知信息在所有事件中的分布比例如图所示，由于缺失信息达到了 80%多，所以认为其包含有用信息的可能性较少，数据处理时将其剔除。下图 4.3 是相应的统计信息。

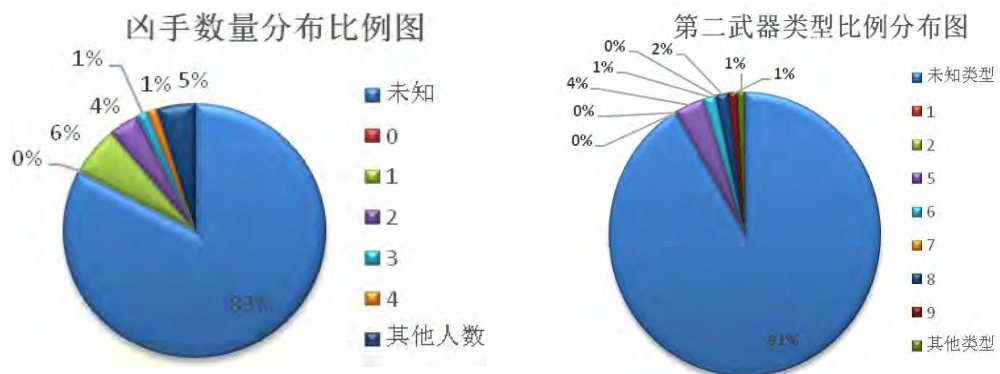


图 4.3 凶手数量、第二武器分布比例统计图

##### (2) 缺失数据填充

对于给出的数据 Excel 附件 1，部分恐怖事件信息存在缺失情况。统计学将数据缺失的记录称为不完全观测，这种数据缺失或不完全观测对调查研究有着很大的影响。

完全随机缺失 (MCAR)，在 MCAR 假设下，数据缺失的原因与观测到的变量和未观测到的变量无关。随机缺失 (MAR)，在 MAR 假设下，数据缺失原因取决于完全观测到的协变量（如干预、基线），而与未观测到的因素无关。非随机缺失 (MNAR)，不完全变量中数据的缺失依赖于不完全变量本身，这种缺失

是不可忽略的。

经分析题目所给恐怖事件数据缺失情况并根据上述类型描述得出，该数据缺失属于随机缺失（MAR）机制，采用 SPSS 数据软件采用最大似然估计算法对缺失数据进行填充。如附件 1 中给出的死亡总数一项，其数据情况比例图如图 4.4 所示，缺失数据占总数的 7%左右，需要对数据进行填补。

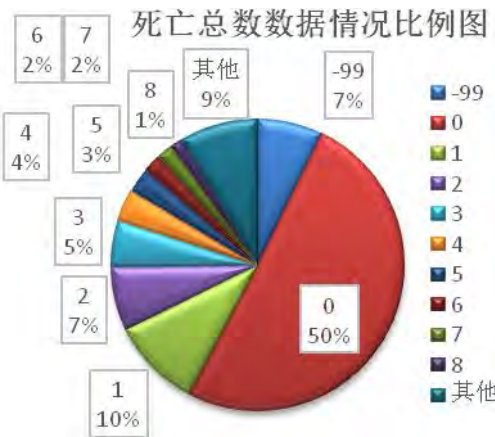


图 4.4 死亡总数情况比例图

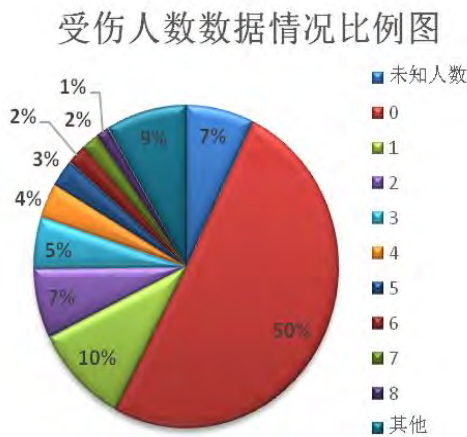


图 4.5 受伤人数情况比例图

4.2.3 数据标准化

分析给出的附件 1 恐怖事件数据发现，恐怖事件不同属性之间的量级存在着较大的差异，如附件 1 中给出的 `nwound`（受伤总数），其数据分布十分广泛，最大到 8191，最小到 0，而且颇大的数据分布较少，部分数据如表 4.2 所示<sup>[3]</sup>。

表 4.2 受伤总数量级比例示意表

受伤人数	频数	所占比例
8191	1	0.00%
8190	1	0.00%
4000	1	0.00%
1500	1	0.00%
1001	1	0.00%
851	1	0.00%
817	1	0.00%

750	2	0.00%
•	•	•
•	•	•
•	•	•
4	4025	3.53%
3	5891	5.16%
2	8049	7.05%
1	11520	10.09%
0	57626	50.47%

这对于后续的聚类处理有着很大的影响,在此本文采用最大最小值方法对部分数量级差别比较大的进行归一化处理,公式如 4.1 所示。其目的是消除各维数据的数量级之间的差别。

$$x_k = (x_k - x_{\min}) / (x_{\max} - x_{\min}) \quad (4.1)$$

通过分析所给的恐怖事件数据,不同恐怖事件属性的正负相关性存在一定差异,如恐怖事件人员伤亡值越大危害性越大,而财产损害程度分级则不同,财产损害程度越高所代表的数值越小,为便于后续的聚类,将每一个属性的正负相关性进行统一,及每一类特征向量数值越大代表该项造成的危害程度越高。

经过数据预处理后,获得的原始危险等级特征向量如表 4.3 所示,处理后的数据共包含有 14 组与恐怖袭击相关的特征向量,但是这些特征向量并非全都与危害等级评价有关,需要对特征向量进行进一步地筛选和降维处理。

表 4.3 危害性影响主要指标

编号	含义	编号	含义
1	武器种类	8	受伤总数
2	地区代码	9	组织声明
3	成功攻击	10	绑架人数
4	攻击类型	11	自杀式袭击
5	受害者类型	12	凶手死亡人数
6	国家代码	13	凶手受伤总数
7	死亡总数	14	财产损失等级

### 4.3 恐怖袭击分级特征向量提取

主成分分析是采取一种数学降维的方法<sup>[4]</sup>,其所要做的就是设法将原来众多具有一定相关性的变量,重新组合为一组新的相互无关的综合变量来代替原来的变量。通常,数学上的处理方法就是将原来的变量做线性组合,作为新的综合变量,期一般流程如图 4.6 所示。

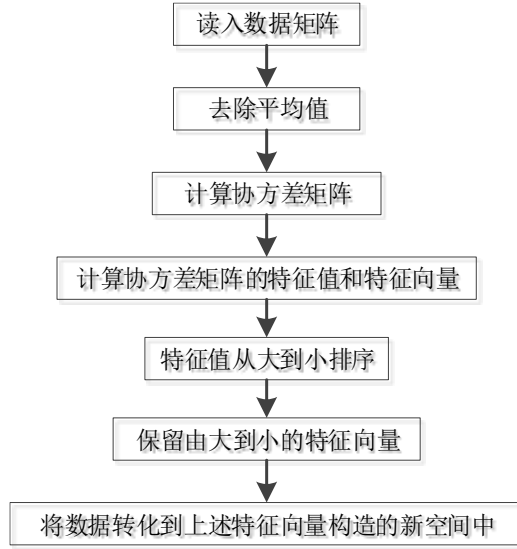


图 4.6 PCA 分析原理图

(1) 对原始数据进行标准化处理后。假设样本的观测矩阵为：

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$

那么按照如下的方法对原始数据进行标准化处理：

$$x_{ij}^* = \frac{x_{ij} - \bar{x}_j}{\sqrt{\text{Var}(x_j)}} \quad (i = 1, 2, \dots, n; j = 1, 2, 3, \dots, p) \quad (4.2)$$

其中，

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij}, \text{Var}(x_j) = \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2 \quad (j = 1, 2, 3, \dots, p) \quad (4.3)$$

(2) 计算样本相关系数矩阵。

为了方便，假定原始数据标准化后仍用  $X$  表示，则经过标准化处理后的数据的相关系数矩阵为

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1p} \\ r_{21} & r_{22} & \cdots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{np} \end{bmatrix}$$

其中，

$$r_{ij} = \frac{\text{Cov}(x_i, x_j)}{\sqrt{\text{Var}(x_i)}\sqrt{\text{Var}(x_j)}} = \frac{\sum_{k=1}^n (x_{ki} - \bar{x}_i)(x_{kj} - \bar{x}_j)}{\sqrt{\sum_{k=1}^n (x_{ki} - \bar{x}_i)^2} \sqrt{\sum_{k=1}^n (x_{kj} - \bar{x}_j)^2}} \quad (4.4)$$

(3) 计算相关系数  $R$  的特征值  $(\lambda_1, \lambda_2, \dots, \lambda_p)$  和相应的特征向量:

$$a_i = (a_{i1}, a_{i2}, \dots, a_{ip}) \quad (i = 1, 2, \dots, p) \quad (4.5)$$

(4) 选择重要的主成分，并写出主成分表达式。

主成分分析可以得到  $p$  个主成分，但是，由于各个主成分的方差是递减的，包含的信息量也是递减的，所以实际分析时，一般不是选取  $p$  个主成分，而是根据各个主成分累计贡献率的大小选取前  $k$  个主成分。这里贡献率是指某个主成分的方差占全部方差的比重，实际也就是某个特征值占全部特征值合计的比重，所以

$$\text{贡献率} = \frac{\lambda_i}{\sum_{i=1}^p \lambda_i} \quad (4.6)$$

贡献率越大，说明该主成分所包含的原始变量的信息越强。主成分个数  $k$  的选取，主要根据主成分的累计贡献率来决定，即一般要求累计贡献率达到 85% 上，这样才能保证综合变量能包括原始变量的绝大多数信息。

(5) 计算主成分得分。

根据标准化的原始数据，按照各个样品，分析带入主成分表达式，就可以得到各主成分下各个样本的新数据，及为主成分。具体形式如下：

$$\begin{bmatrix} F_{11} & F_{12} & \dots & F_{1k} \\ F_{21} & F_{22} & \dots & F_{2k} \\ \dots & \dots & \dots & \dots \\ F_{n1} & F_{n2} & \dots & F_{nk} \end{bmatrix}$$

(6) 依据主成分得分的数据，进一步对问题进行后续分析与建模，后续分析与建模通常的形式有主成分回归，变量子集合的选择，综合评价等。

依照上述建立的主成分分析模型，求解个主成分向量所对应的特征值，贡献率，以及累计贡献率如表 4.4 所示。

表 4.4 各主成分特征值与贡献率

特征值	贡献率	累计贡献率
5.7361	0.6421	0.6421
1.0972	0.2385	0.8806
0.5896	0.0472	0.9278
0.2858	0.0256	0.9534
0.1456	0.0135	0.9669
0.1369	0.0092	0.9761
0.1256	0.0088	0.9849
0.0945	0.0066	0.9915
0.0698	0.0034	0.9949
0.0365	0.0022	0.9971
0.0268	0.0011	0.9982
0.0189	0.0009	0.9991
0.0105	0.0006	0.9997

0.0098	0.0003	1
--------	--------	---

选取主成分特征保留率  $T=0.9$ , 则求得的主成分数为 3, 即保留贡献率最高的前三类主成分, 其对应的特征向量如表 4.5 所示

表 4.5 保留率为 0.9 时提取的特征向量

特征向量 1	特征值向量 2	特征值向量 3
0.4155	-0.144	-0.2886
0.3957	-0.3925	0.0601
0.3148	0.5782	-0.4523
0.3745	-0.0163	0.6274
0.3599	0.4866	0.1241
0.6689	0.1021	0.5943
0.6638	0.6065	-0.1222
0.4844	0.6591	0.2938
0.6041	-0.5008	0.1323
0.3539	0.5498	-0.168
0.4603	0.5408	0.6765
0.648	0.3232	0.5864
0.3872	0.0864	0.322
0.6647	-0.0861	0.7652

由于各主成分是原向量的线性组合, 根据主成分向量的贡献率可以反过求原向量的贡献权重, 各个原向量权重如表 4.6 所示:

表 4.6 各原向量权重分布

编号	含义	贡献权重	编号	含义	贡献权重
1	武器种类	0.06	8	受伤总数	0.11
2	地区代码	0.21	9	组织声明	0.005
3	成功攻击	0.03	10	绑架人数	0.005
4	攻击类型	0.03	11	自杀式袭击	0.01
5	财产损失等级	0.1	12	凶手死亡人数	0.04
6	国家代码	0.01	13	凶手受伤总数	0.04
7	死亡总数	0.22	14	受害者类型	0.13

各向量之间的相关性强弱分布如图 4.7 所示, 其中对脚线元素表示自相关, 其他位置的元素表示互相关, 颜色越深, 接近于黄色表明相关性越强, 反之相关性越弱。

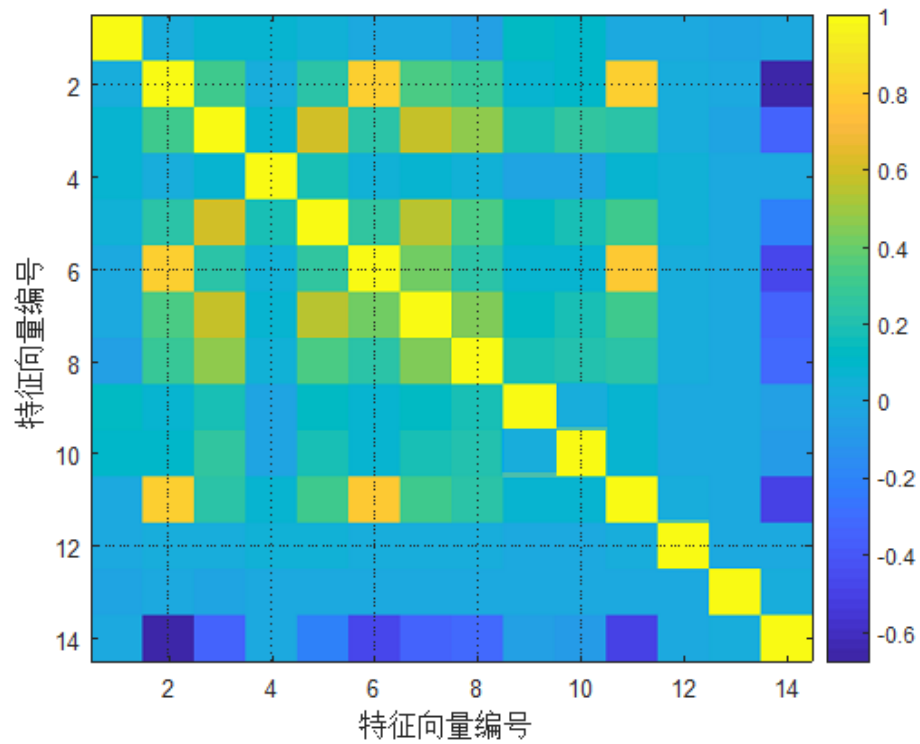


图 4.7 原向量关联强度图

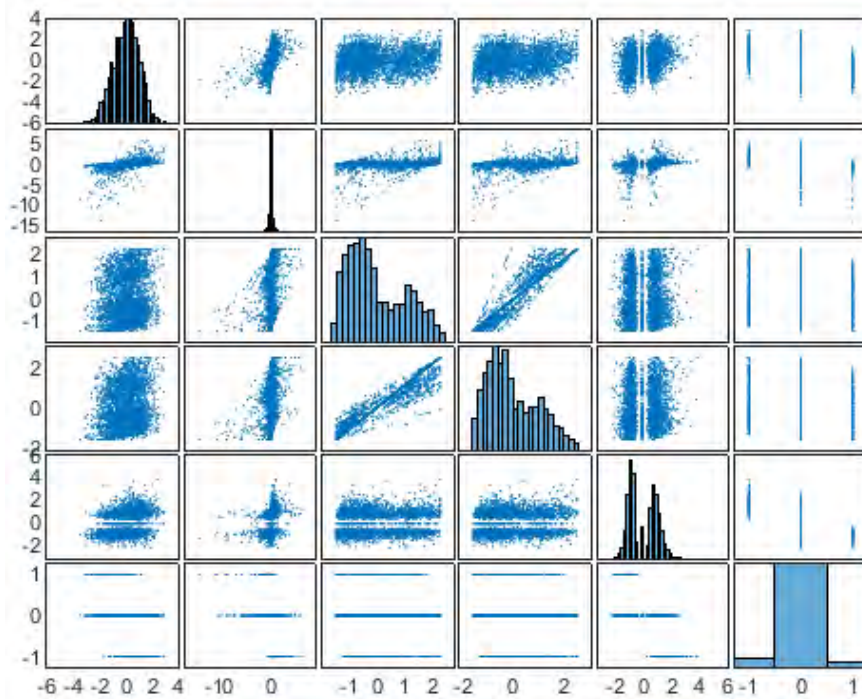


图 4.8 部分变量间相关性关联图

#### 4.4 聚类分析-事件分级

本文进行聚类分析对恐怖主义事件进行分级的总体思路如下图 4.9 所示。



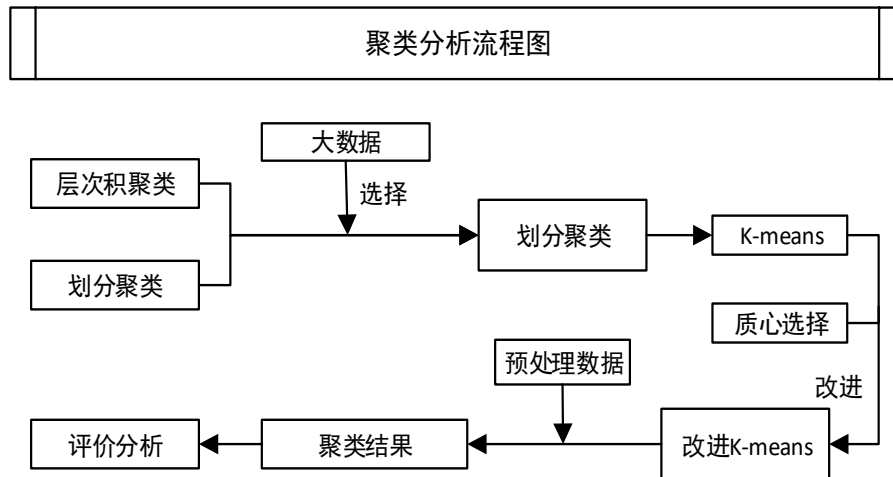


图 4.9 聚类分析流程图

#### 4.4.1 聚类分析方法调研

针对问题一的要求，需要对近二十年的恐怖事件进行分级，聚类分析方法在大数据的挖掘处理过程中起着重要的作用。目前主流的聚类分析方法有划分聚类、层次聚类、基于密度的聚类、基于网格的聚类。对这些聚类方法的特点和适用范围进行调研，得到结果如下表 4.6 所示。

表 4.6 聚类分析比较

聚类方法	适用范围	优点	缺点
划分聚类	中等体量数量集	对于大型数据集也是简单高效、时间复杂度、空间复杂度低	结果容易局部最优；需预设 K 值，且对 K 点个数选取敏感；对噪声和离群值敏感。
层次聚类	小数量集	可解释性好；能产生高质量的聚类；可以用于非球形族	时间复杂度高。
基于密度的聚类	任意族形状	对噪声不敏感；能发现任意形状的聚类。	聚类的结果与参数关系紧密；较稀的聚类或离得较近的类效果不好。
基于网格的聚类	底层数据密度小	速度很快	参数敏感、无法处理不规则分布的数据、维数灾难等；算法效率以聚类结果的精确性为代价

由上表可知，划分聚类和层次聚类应用较为普遍，但是层次聚类的计算复杂度是  $O(n^2)$ ，对于本文对 11 万个样本的处理并不适合，因此本文选择 K-means 聚类方法来对数据进行分析。K-means 方法的计算复杂度是  $O(n)$ ，适合于大数据

的处理分析。但 k-means 方法存在以下缺点：首先，需要事先给定 K-means 算法中，但 K 值的选定通常是不确定的。很多时候，事先并不知道给定的数据集应该分成多少个类别才最合适；其次在 K-means 算法中，首先需要随机选择初始聚类质心，然后不断进行迭代计算。这个初始聚类中心的选择对聚类结果影响较大，一旦初始质心选择的不好，可能无法得到有效的聚类结果。

#### 4.4.2 改进的 k-means 算法

为了克服传统 K-means 的缺点，本文提出了一种改进的 K-means 算法。由于题目要求规定需要把所有恐怖袭击事件分成 5 个等级，即 k 值没法进行优化，所以本文主要针对初始聚类质心的选择进行优化。改进 k-means 的核心思想是选择尽可能远的聚类质心点，具体算法原理如下流程框图 4.10 所示。

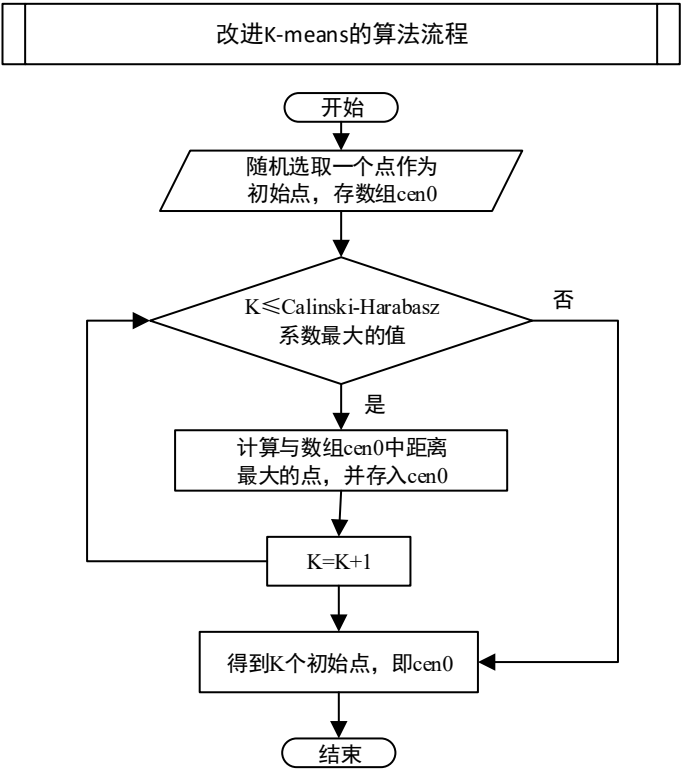
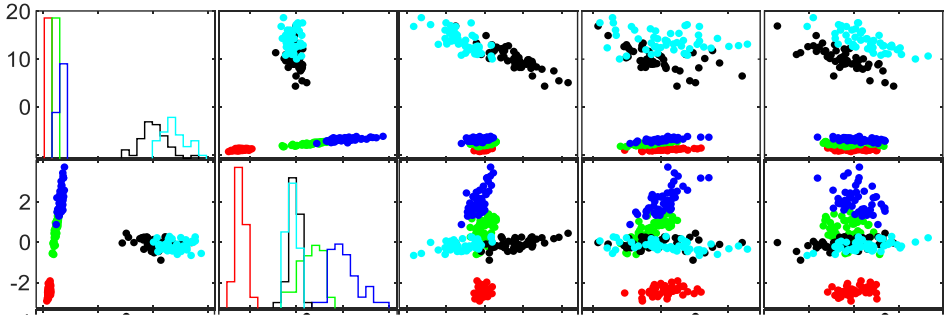


图 4.10 改进 k-means 算法流程图

#### 4.4.3 聚类结果求解

确定了 k-means 的改进算法后，应用 Matlab 对预处理后的数据进行运算，得到恐怖事件的聚类结果如下图 4.11 和 4.12 所示。



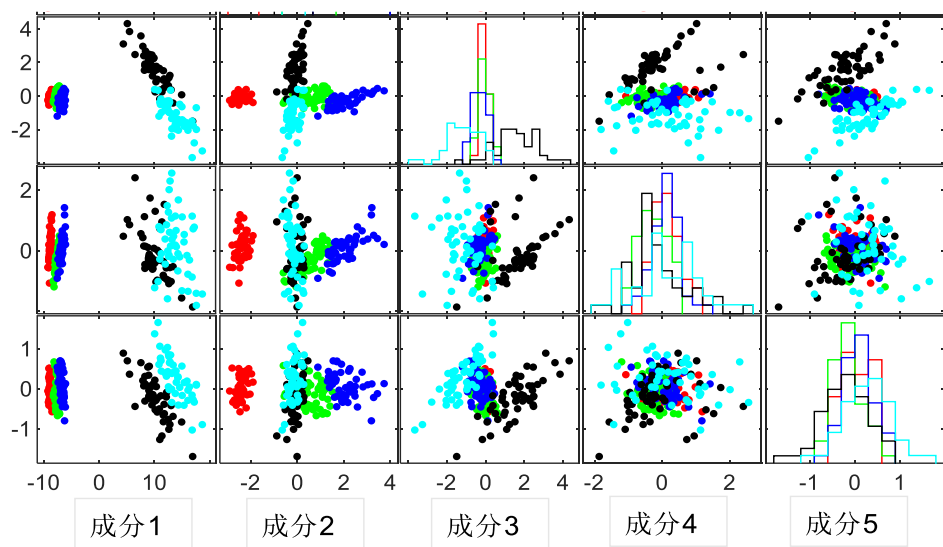


图 4.11 不同成分聚类结果 1

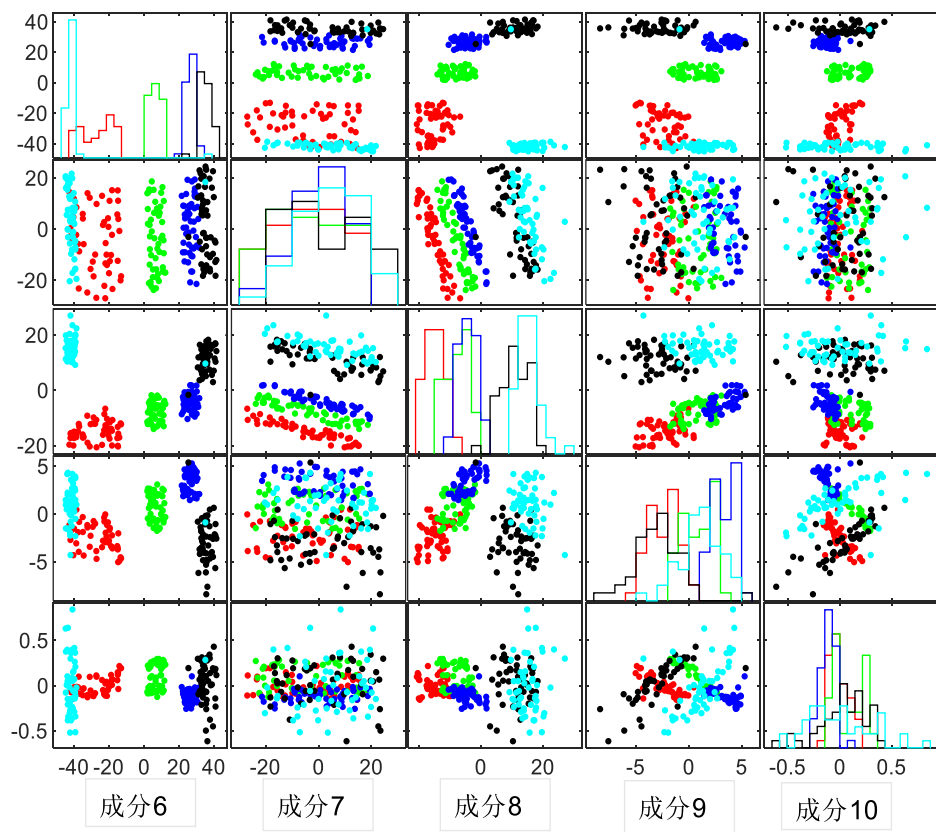


图 4.12 不同成分聚类结果 2

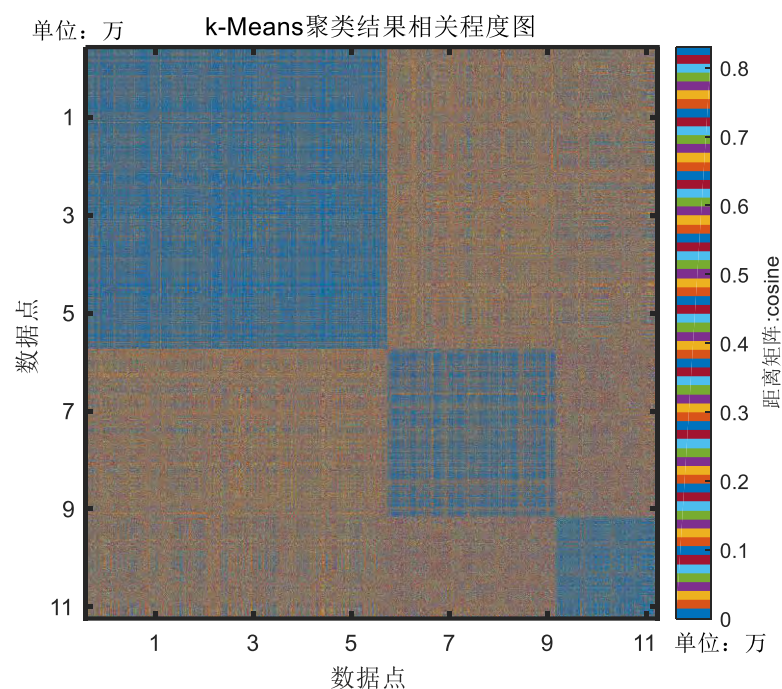


图 4.13 聚类结果相关程度图

根据聚类结果图和聚类结果相关程度图，可以将所有事件分为 5 级，每类事件的数量和比列如下图 4.14 所示。由图可知，1、2、3 级事件的数量较少，大多数事件是 4 级和 5 级。

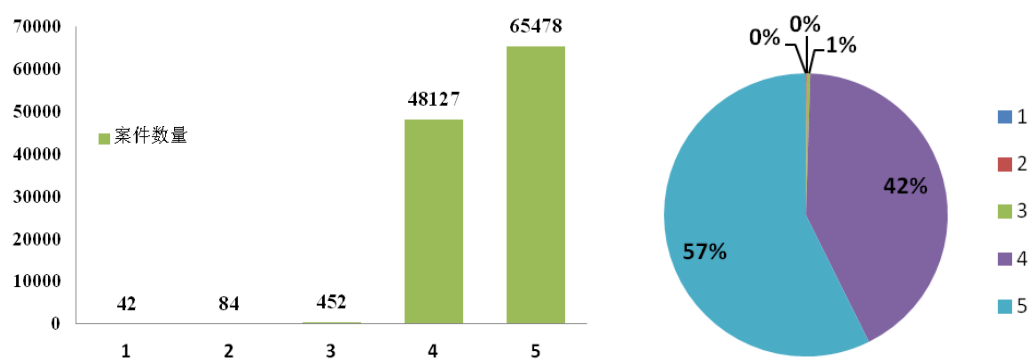


图 4.14 五级事件统计分析

根据聚类结果，找出第一问给出表中事件编号所在的类，从而可以决断各个事件所在的级别，具体结果如下表 4.7 所示。

表 4.7 问题给定事件危害级别判定

事件编号	危害级别
200108110012	2
200511180002	3
200901170021	1
201402110015	4
201405010071	4
201411070002	5
201412160041	2

201508010015	5
201705080012	3

#### 4.5 主成分分析排序-前十恐怖袭击事件

根据上一问聚类分析结果，可以得到分级结果为 1 级的恐怖事件样本，由于问题一还要求找出近二十年的十大恐怖主义事件，因此本文对 1 级恐怖主义事件样本进行主成分分析，把综合得分排名前十的恐怖主义袭击事件定义为近二十年十大恐怖主义袭击事件。

主成分分析进行排序找出近二十年十大恐怖主义袭击事件的思路如下图 4.15 所示。

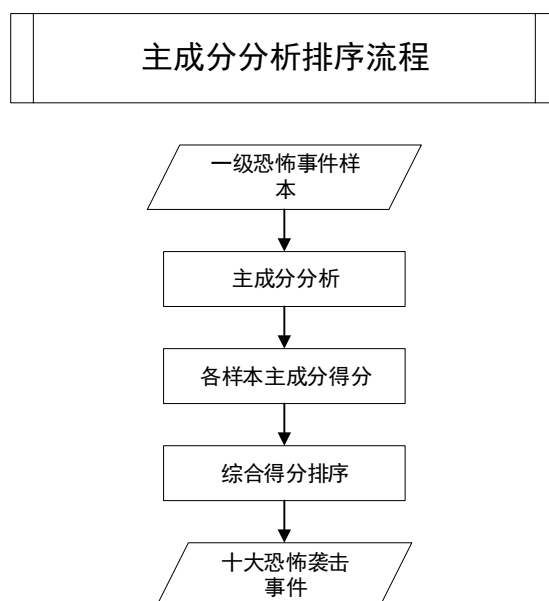


图 4.15 主成分分析排序流程图

将有多个事件编号但是同一恐怖事件性质的事件算作一个编号，应用主成分分析对 1 级事件样本进行计算，并将总得分进行排序，结果如下表 4.8 和表 4.9 所示。

表 4.8 各主成分特征值与贡献率

编号	特征值	方差贡献率	累积贡献率
1	8.077464583	0.807746458	0.807746458
2	1.163279864	0.116327986	0.924074445
3	0.295368546	0.029536855	0.953611299
4	0.256892667	0.025689267	0.979300566
5	0.178136584	0.017813658	0.997114224
6	0.023679934	0.002367993	0.999482218
7	0.003121836	0.000312184	0.999794401
8	0.001321209	0.000132121	0.999926522
9	0.000580407	5.80E-05	0.999984563
10	0.000154372	1.54E-05	1

表 4.9 主成分指标系数

	主成分 1	主成分 2
x1	0.34459435	0.175508396
x2	0.345546719	0.032157799
x3	0.348876537	0.0497364
x4	0.348499799	0.068664534
x5	0.347131475	0.113310445
x6	0.346897866	0.12447487
x7	0.286616629	-0.293533541
x8	0.079055886	-0.874969753
x9	0.313708152	1.16E-01
x10	0.301720285	-2.60E-01

表 4.10 主成分分析综合得分排序

编号	prim1 得分	prim2 得分	综合得分	排序
1	9.900297377	1.0581777	10.95847508	1
2	4.723495394	-0.707633489	4.015861906	2
3	2.640855992	-1.000441277	1.640414716	3
4	0.068826133	1.195050825	1.263876957	4
5	-1.672843977	2.065452889	0.392608912	5
6	0.194566464	0.197086009	0.391652474	6
7	-1.599656624	1.465347014	-0.13430961	7
8	0.164129479	-0.466069714	-0.301940235	8
9	-0.264626091	-0.042702846	-0.307328937	9
10	-1.846308732	0.981899975	-0.864408758	10
11	-1.534160549	0.618704972	-0.915455577	11
12	-0.496531256	-0.506715123	-1.003246379	12
13	-1.862712917	0.739165278	-1.123547638	13
14	-1.395617733	0.142105103	-1.25351263	14
15	-1.369960441	-0.076887768	-1.446848209	15
16	-1.84248506	0.276806776	-1.565678284	16
17	-0.752842274	-1.013638903	-1.766481177	17
18	-0.582462395	-1.711377228	-2.293839623	18
19	-1.151808713	-1.55E+00	-2.701674474	19
20	-1.320154079	-1.66E+00	-2.984618511	20

根据上表 4.10 主成分分析综合得分排序，找出近二十年危害性排名前十大恐怖袭击事件如下表 4.11 所示。

表 4.11 十大恐怖袭击事件

事件	序号
1	200109110004
	200109110005
	200109110006
	200109110007
2	200409010002
3	200210230004
4	200507070001
	200507070002
	200507070003
	200507070004
5	200811260001
	200811260002
	200811260003
6	201710140002
	201710140003
7	199808070002
	199808070003
8	200403110001
	200403110003
	200403110004
	200403110005
	200403110006
	200403110007
9	200210120004
	200210120005
	200210120006
10	201101240016

## 4.6 模型检验

### 4.6.1 聚类结果评价

对于改进的 k-means 算法进行关于聚类质量和有效性分析,通常可以分为有基准和无基准的情况。本文没有基准故采用内在方法进行评估。

评价 k-means 聚类结果的参数有很多,邓恩系数(DVI),轮廓系数(Silhouette)。各个参数指标的计算公式如下所示<sup>[5]</sup>。

$$DVI = \frac{\min_{0 < m \neq n < K} \left\{ \min_{\substack{\forall x_i \in \Omega_m \\ \forall x_j \in \Omega_n}} \{ \|x_i - x_j\| \} \right\}}{\max_{0 < m \leq K} \max_{\forall x_i, x_j \in \Omega_m} \{ \|x_i - x_j\| \}} \quad (4.7)$$

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (4.8)$$

DVI 越大意味着类间距离越大 同时类内距离越小；S (i) 值越小，反簇越紧凑，值越大，则与其簇越分离。利用 Matlab 计算出两个参数如下表 4.12 所示。

表 4.12 聚类参数评价结果

算法	DVI	S(i)
K-means	0.03625616	0.3616042
改进的 K-means	0.04261861	0.3950412

由上表评价参数可知，本文采用的改进 K-means 算法聚类得到的结果簇内相似度较高，簇间相异度明显，说明聚类效果良好

#### 4.6.2 相关性检验

由上表所示，本文提出的聚类结果簇内相似度很高，簇间相异性很大，聚类结果较好。

本文模型把 11 万多个恐怖主义事件的样本分成了 5 个等级，利用复相关系数来对模型进行检验。复相关系数反映一个因变量与多个自变量的相关性。

$$(4.9)$$

$$(4.10)$$

利用 matlab 计算出本文模型聚类出的恐怖事件等级与相应指标的相关系数为

$$R=0.7233$$

由相关性分析说明本文对恐怖事件等级划分的指标选取可靠性较高，获得聚类结果具有较高的精确度

## 五、问题二的模型建立与求解

### 5.1 问题分析

根据任务要求得出，该问题数据源范围为 2015-2016 年，尚未有组织或个人宣称负责，并且未知犯罪集团的恐怖袭击事件。分析数据信息，其中每个恐怖事件包含了年月日编号、年、月、日、大概日期、国家代码、国家名称、死亡人数等诸多事件属性信息，经查阅文献，事件数据信息具有较大的数据冗余，并且部



分属性信息采用文字描述，在结构规范性上存在一定缺陷。

需要完成的具体任务为，运用数学模型将不同时间、不同地点多次作案的若干案件寻找将其归为一类的可能性，并对其所对应的未知作案组织或个人进行标识，然后根据其危害性进行排序，并预测列表中给出的部分 2017 年恐怖事件归为哪个组织的嫌疑度大小。经调研相关文献信息，了解了常用的事件关联方法，挖掘数据源信息，采用逻辑回归等方法对事件进行模型构建，并利用第一问危害性分级指标对未知作案组织或个人进行排序，并依据所给列表中事件属性进行嫌疑度排序。

确定任务二的总体思路如图 5.1 所示。

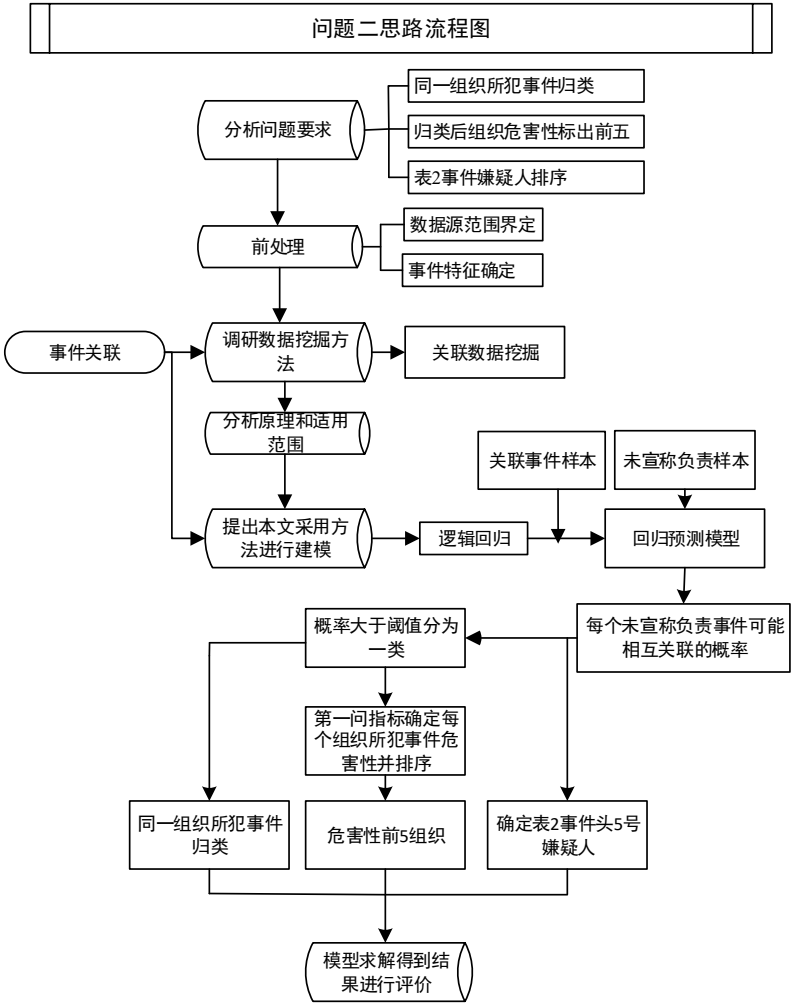


图 5.1 问题二思路流程图

### 5.2 数据处理

根据任务要求，确定数据源范围，剔除冗余的数据，对数据进行降维处理，筛选出用于恐怖事件关联规则挖掘的信息，并对数据中文字描述的特征属性进行结构上的规范性统一，数据处理流程图如下图 5.2 所示。

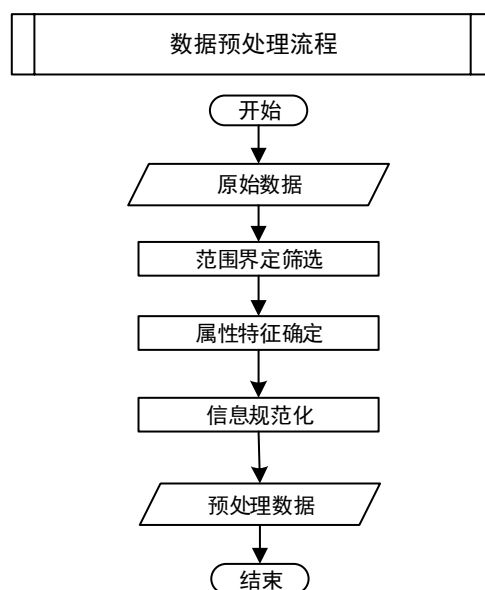


图 5.2 数据预处理流程

### 5.2.1 范围界定筛选

根据任务要求，针对在 2015、2016 年度发生的、尚未有组织或个人宣称负责的恐怖袭击事件，将其他年份以及已有组织或个人宣称负责的恐怖袭击事件以及未有组织或个人宣称负责但是能查到对应犯罪集团的样本剔除，确定本次任务数据源对象范围。

### 5.2.2 属性特征确定

恐怖袭击事件通常会带有时间特征、空间特征以及特有属性特征等<sup>[6]</sup>，通常用于关联分析的恐怖袭击事件数据属性应包含：

- (1) 时间特征：恐怖袭击时间发生的时间，可能是时间段，也可能是时间节点。
- (2) 空间特征：恐怖袭击时间发生的地点，根据不同数据源的精度。位置信息会有区域、国家和地区，以及经纬度坐标等不同的表示形式。
- (3) 属性特性：恐怖袭击事件的类型、受害者类型、死亡人数、受伤人数、武器类型以及攻击类型等。

任务信息所给的附件 1 源于 GTD 数据库，较为完善的记录了恐怖袭击事件的各方面信息，包含一百多条恐怖事件的属性特征项，但绝大多数的事件特征属性项与事件核心关联之间没有很强的关联性，不能够在进一步的数据挖掘过程中起到一定的作用，因此该数据源信息具有较大的数据冗余。

经查阅相关文献，结合上述恐怖袭击事件的时间特征、空间特征以及属性特征，考虑到本文的研究任务需要研究恐怖袭击事件内部核心要素与恐怖袭击事件之间时间和空间之间的关联关系，以任务所给数据源信息为基础，结合关联分析所需的要素，确定保留了事件相关联的核心关联要素特征，属性特征信息如表 5.1 所示。

表 5.1 属性特征信息表

名称	含义
imonth	恐怖事件发生的月份
region	恐怖事件发生的地区

weaptype1	武器类型
targtype1	受害者类型
attacktype1	攻击类型

### 5.3.3 信息规范化

分析保留的恐怖袭击事件特征属性信息，部分信息是由文字进行描述，为了简化计算，节省内存空间，参照附件 2 的 GTD 全球恐怖主义数据库变量说明，对特征属性信息进行数字编码对照表示，部分对照信息如表 5.2 所示。

表 5.2 部分属性编号

region	编码
North America	1
Middle East & North Africa	2
South America	3
East Asia	4
•	•
•	•
•	•

经上述流程，对任务给出的恐怖袭击事件数据源进行处理后，得到最后用于恐怖袭击事件关联信息挖掘的部分事件数据信息表如表 5.3 所示。

表 5.3 五大关联特征

事件编号	imonth	region	weaptype1	targtype1	attacktype1
201412030034	1	10	6	14	3
201412220095	1	9	9	15	2
201501010001	1	10	6	8	3
201501010002	1	8	8	15	7
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•

## 5.3 恐怖主义事件关联规则挖掘

### 5.3.1 恐怖主义事件特征统计分析

恐怖主义事件在时间、空间及其它作案特征上具有一定的倾向性，利用统计学分析手段可以表现事件发生频数、事件的空间属性和专题属性值在不同时间对比与变化趋势，可以表现事件专题属性值在不同空间范围内的对比与变化趋势。既可以对事件整体变化进行模拟也可以对某一属性在时空范围内进行跟踪<sup>[6]</sup>。

在数据处理过程中，本文根据参考文献定义出了能反映同一恐怖组织犯案的特征的 5 个属性，为了对这些事件特征属性的分布有清晰的统计学认识，主要利用 excel 和 matlab 对这些数据进行统计。样本中的恐怖主义事件空间分布特征如下图 5.3 所示。



图 5.3 恐怖主义事件全球空间分布特征

由上图可知，在 2015 到 2016 年，恐怖主义事件显示出明显的地域特性，在中东地区、北非、南亚、东南亚地域是明显的恐怖主义多发地区，而在美洲地区则明显较少，同时能发现欧洲地区的恐怖主义袭击事件态势也比较严峻。

恐怖主义事件还与时间、武器类型、事件攻击类型、事件受害者等其它特征密切相关，如下图 5.4 所示。



图 5.4 恐怖主义事件特征分布图

上面的横坐标轴按 GTD 数据变量说明中数字进行定义的，由上图可知，恐怖分子选择轻武器和爆炸物进行袭击占大多数，在月份分布上基本均匀，攻击对象则更多的是公民自身和财产。

通过对恐怖事件的特征进行统计分析,可以为未宣称负责的恐怖主义袭击事件寻找可能犯案的恐怖主义组织或个人提供更直观的分析。

### 5.3.2 逻辑回归事件相似度计算

逻辑回归属于概率型非线性回归<sup>[7]</sup>,它可以研究多个变量之间的关系,适用于本问要求的对未宣称负责的恐怖主义事件的分类。

本文因变量是一个二项分类变量,即两个事件是否关联是否由同一个人或同一伙所为,而自变量都是区间标度变量。本文事件的属性都是区间标度变量,如地区编号是 1-12。

对于区间标度变量的相似度计算,本文采用的公式是

$$S_i(A,B)=1-\frac{|v_i(A)-v_i(B)|}{R_i} \quad \text{式 (5.1)}$$

式中,A、B 表示不同的恐怖主义袭击事件,i 表示事件的某一特征属性,  $v_i(A)$  和  $v_i(B)$  表示不同事件 A、B 在同特征属性上的值,  $R_i$  表示事件该属性的取值范围,这样可以保证将计算得到的相似度值定义在 0 到 1 之间,从而使关联结果受特征属性值得量纲的影响较小。

在本文中,两个事件的关联概率逻辑回归模型为

$$p(linked)=\frac{e^{\alpha+\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_nX_n}}{1+e^{\alpha+\beta_1X_1+\beta_2X_2+\beta_3X_3+\beta_nX_n}} \quad \text{式 (5.2)}$$

P(linked) 表示两个事件相互关联的概率,  $\alpha$  表示常量,  $\beta_i$  表示逻辑回归系数,  $X_n$  表示事件各类属性的相似度。计算得到的概率越大,则两个事件的关联性越大<sup>[8]</sup>。

通过数据处理中得到的 2015/2016 年的数据,发现部分事件确定有相应的关联事件,利用这些事件来计算逻辑回归模型的参数,故提取出 5 对未宣称负责事件的关联事件下表 5.4 所示。

表 5.4 部分相似关联数据对

	imonth	region	weaptype1	targtype1	attacktype1
相似关联数据对 1	1	10	3	1	6
相似关联数据对 2	1	9	3	1	6
相似关联数据对 3	2	6	3	14	6
相似关联数据对 4	1	9	3	14	6
相似关联数据对 5	2	6	3	14	6
相似关联数据对 6	1	9	3	14	6
相似关联数据对 7	6	5	7	8	8
相似关联数据对 8	1	8	7	8	8
相似关联数据对 9	6	5	7	8	8
相似关联数据对 10	1	8	7	8	8

根据每个属性特征的计算公式,计算出各对关联事件的相似度如下表 5.5 所示,从而为后续用 SPSS 软件进行分析准备。

表 5.5 部分数据对各指标相似度

数据对名称	imonth	region	weaptype1	targetype1	attacktype1
相似关联数据对 1	1	0.91667	1	1	1
相似关联数据对 2	0.91667	0.75	1	1	1
相似关联数据对 3	0.91667	0.75	1	1	1
相似关联数据对 4	0.58333	0.75	1	1	1
相似关联数据对 5	0.58333	0.75	1	1	1

### 5.3.3 逻辑回归预测模型确立

根据上文计算出的 5 对关联事件利用 SPSS 软件对上述关联事件做逻辑回归分析，得到结果如下表 5.6 所示。

表 5.6 逻辑回归分析结果一

		B	S. E.	wald	df	sig
step1	月份	1.356	2.198	6.123	1	0.023
	地区	2.135	1.697	6.696	1	0.013
	武器类型	3.269	4.216	5.462	1	0.008
	受害者类型	5.236	1.930	9.757	1	0.002
	攻击类型	4.187	3.564	2.868	1	0.019
	constant	-8.330	1.806	6.162	1	0.002

根据逻辑回归分析图一，各个特征的 sig 值都小于 0.05，因此认为事件关联与选定的五个特征相似度具有统计学意义，可以纳入逻辑回归预测模型方程。

表 5.7 逻辑回归分析结果二

step	-2 lg likelihood	Cox&Snell R Square	Nagelkerke R Square
1	15.062	0.653	0.856

逻辑回归分析结果一显示了模型拟合优度方面的测试指标。如表 5.7 所示，最终模型的-2 倍对数似然函数值为 15.062,比较低,说明模型的拟合优度比较理想。同时 R<sup>2</sup> 距 1 较近，也说明模型的拟合优度较高。

由以上分析结果可确立逻辑回归预测模型为：

$$p(linked) = \frac{e^{-8.330+1.356X_1+2.135X_2+3.269X_3+5.236X_4+4.187X_5}}{1+e^{-8.330+1.356X_1+2.135X_2+3.269X_3+5.236X_4+4.187X_5}}$$

### 5.3.4 利用逻辑回归预测模型进行求解

根据题目要求，对于预处理后的数据样本，首先将可能是同一组织或个人所犯的恐怖事件归为一类，本文具体思路如下图 5.5 所示。

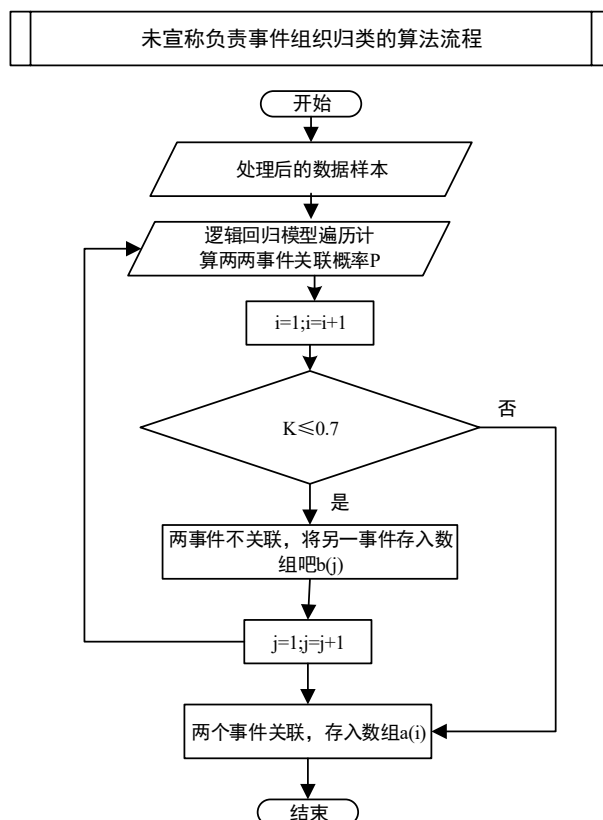


图 5.5 未宣称负责事件组织归类流程图

根据前面获得的逻辑回归预测模型，可以计算出任意两个事件的关联概率，设定当概率大于 0.7 时则认为这两个事件是同一组织所犯下的，将这两个事件归为一类，事先选定一个事件，用逻辑回归预测模型将它与所有事件进行比较，可以得到可能与它是同一组织或个人犯下所有可能案件归为一类，对于概率小于 0.7 事件，接下来重复上述操作，即可以将所有可能是同一组织或个人所犯事件进行归类。

利用 Matlab 编程实现算法，得到结果并进行统计分析如下图 5.6 所示。

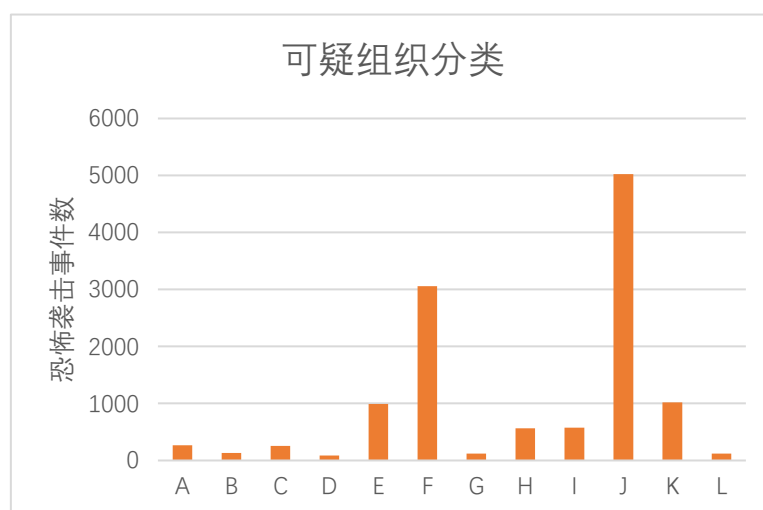


图 5.6 可疑组织分类

根据逻辑回归预测模型，将所有未宣称负责的事件分为了 A-L 类，并且统计出了每一组织犯下的事件数。

对所有未宣称负责的恐怖主义事件进行归类后，需要对各个分类后的恐怖组织进行按危害性进行排序并标记前 5 个，本文利用第一问主成分分析得到的指标对每一类组织犯下的事件进行综合得分排序。最后筛选出得分排序前 5 号结果如下表 5.8 所示。

表 5.8 组织危害性前 5 排序

等级	组织
1	J
2	K
3	F
4	E
5	I

针对要求对第二问表中给出的 2017 年恐怖主义事件寻找它们的可能嫌疑人，并对嫌疑度进行排序，由于逻辑回归预测模型计算出了事件关联的概率，因此可以根据概率大小对嫌疑度进行排序，最后得到结果如下表 5.9 所示。

表 5.9 恐怖分子关于典型事件的嫌疑度

	1 号嫌疑人	2 号嫌疑人	3 号嫌疑人	4 号嫌疑人	5 号嫌疑人
样例 XX	4	3	1	2	5
201701090031	1	2	4	3	5
201702210037	3	1	4	2	5
201703120023	2	1	3	5	4
201705050009	2	1	3	4	5
201705050010	2	1	3	4	5
201707010028	2	3	1	4	
201707020006	1	2	3	5	
201708110018	3	1	2	5	4
201711010006	3	1	2	4	5
201712010003	1	2	3		

5.4 模型评价

本问建立的数学模型关键在于逻辑回归的预测模型的确立。因此在此对逻辑回归预测模型得到的结果进行 ROC 曲线分析<sup>[9]</sup>。结果如下图 5.7 所示。



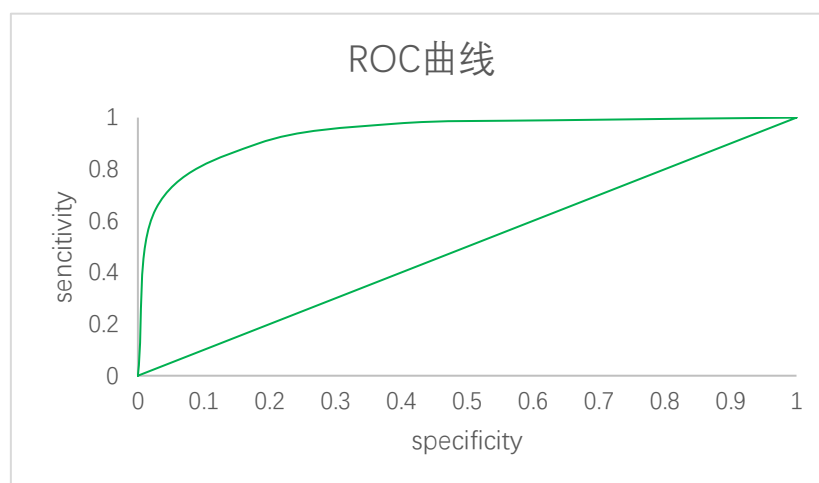


图 5.7 ROC 曲线

由下表可知,曲线下面积为 0.968, (95%的置信区间为 (0.957,1.012)), 标准误差是 0.014, P 值 (Sig.) 为 0.000, 这表示本文所提出的模型准确度较高。

表 5.10 ROC 曲线下面积等有关指标

Area	Std. Error(a)	Asymptotic Sig, (b)	Asymptotic 95% Confidence Interval	
			lower Bound	upper Bound
0.968	0.014	0.000	0.957	1.012

## 六、问题三的模型建立与求解

### 6.1 问题分析

第三问要求对近三年的数据进行分析并建立相应的数学模型, 研究近三年来 (2015 年——2017 年) 恐怖袭击事件发生的主要原因、时空特性、蔓延特性、级别分布等规律, 进而分析判断下一年全球或某些重点地区的反恐态势。

解决这个问题分两步进行, 首先对各个特征向量三年以来的变化规律进行可视化统计分析, 从而确定稳定因素和变化因素, 对变化因素采用灰色预测的方法建立预测模型, 对未来变化趋势和规律进行预测。最后给出反恐斗争的见解和建议。

### 6.2 特征可视化统计分析

首先根据附件提供的数据对 2015 年—2017 年的案件数量, 死亡和受伤人数进行统计, 结果如表 6.1 所示。由表 6.1 统计结果可知, 全球范围内恐怖事件的数量呈明显下降趋势, 由此而带来的伤亡也大大降低, 全球恐怖袭击的状况逐年得到了改善, 这是大的全球趋势和变化规律。

表 6.1 案件结果统计

	2015	2016	2017
案件数量	14963	13592	10897
死亡	38861	34887	26429
受伤	44069	40001	24927

为了充分研究 2015 年—2017 年三年恐怖袭击事件发生的原因，时空特性、蔓延特性、等级分布，我们重点选取了受害者类型，攻击类型，武器类型、地区特征以及结合以及第一问和第二问相关结论进行研究。

### 6.2.1 主要原因分析

恐怖袭击的主要原因主要可以通过受害者类型和攻击类型进行反应。为了方便数据统计和规律观察，每年的数据按照季度进行统计，三年共获得 12 组统计数据，涵盖了 22 种受害者类型，如图 6.1 所示。由于全球反恐环境的改善，全球的恐怖袭击案件逐年减少，为了防止统计规律受总体趋势的影响，又对各受害者类型的比例分布进行了统计，如图 6.2 所示。

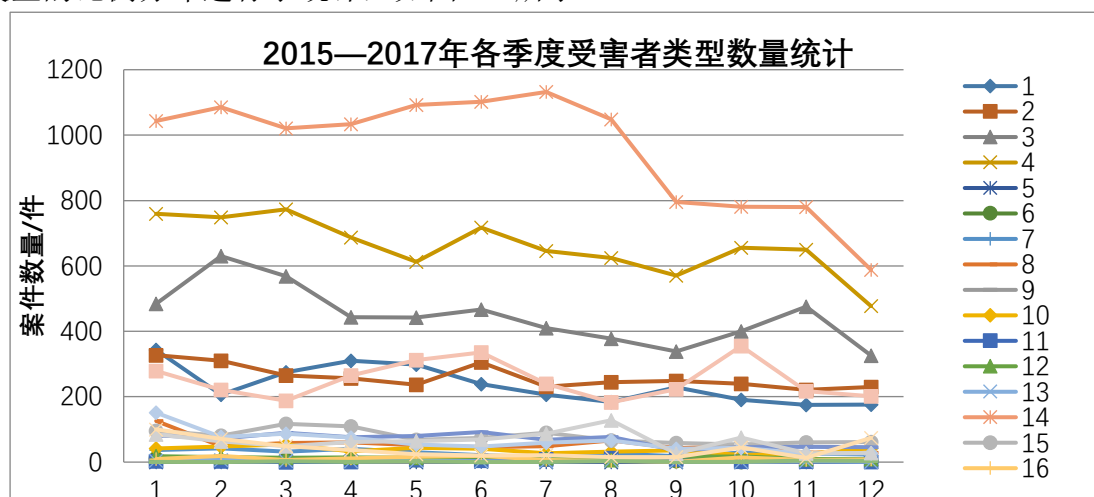


图 6.1 2015—2017 各季度受害者类型数量统计

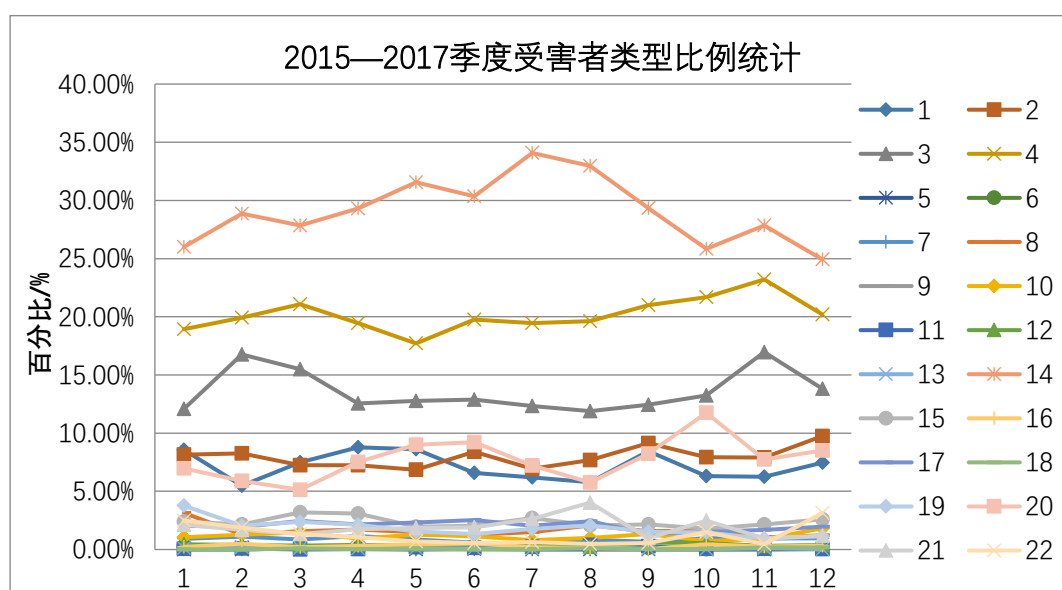


图 6.2 2015—2017 各季度受害者比例统计

由图 6.1 和图 6.2 分析可知，各受害者类型发生案件的数量均呈下降趋势，其中在受害者类型中排名最靠前的前三位分别是宗教（14）、军事（4）、警察（3），在比例分布方面除了宗教（14）、军事（4）、警察（3）有不同程度的波动，其他的袭击受害者类型比例相对稳定，也就是说恐怖袭击这三类占了绝大多数的比重。引起恐怖袭击的原因多数来源与宗教冲突、军事战争等。针对于不同的受害者类型，其攻击类型统计结果如图 6.3 和图 6.4 所示：

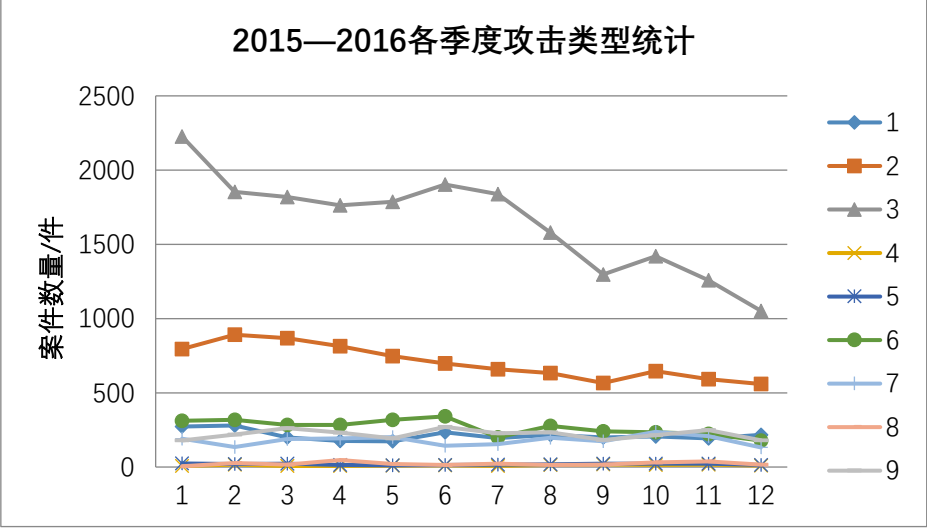


图 6.3 各攻击类型统计

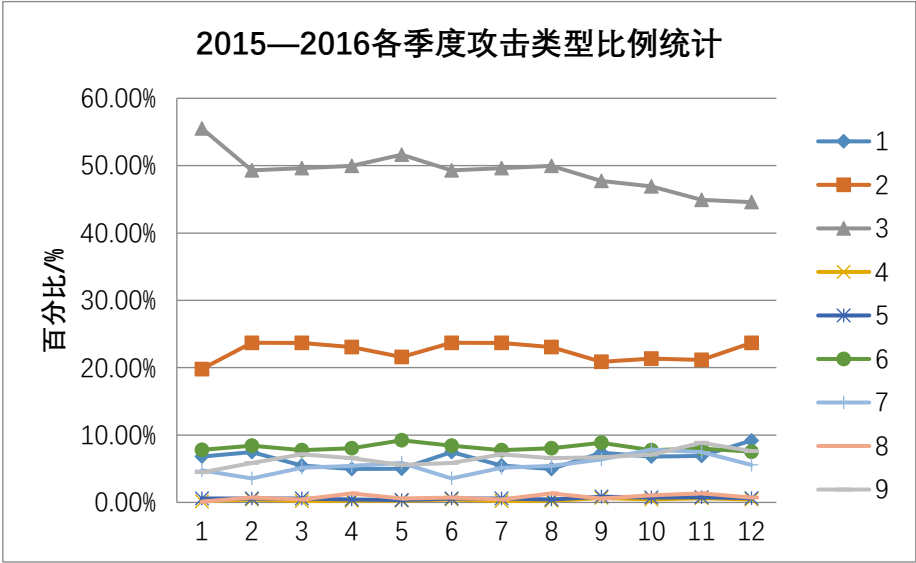


图 6.4 各攻击类型比例统计

由图 6.3 和图 6.4 可得恐怖袭击的攻击类型主要集中于爆炸轰炸（3），武装袭击（2），且各种攻击类型分布比例也相对稳定，也就是说各类恐怖组织实施恐怖袭击的手段比较常规和固定，没有较大的变化。且武器类型的统计结果如表 6.2 所示，其中轻武器（5）和爆炸物/炸弹、炸药（6）约占了总武器攻击类型的 80%。

表 6.2 各季度恐怖袭击武器类型统计

2015			2016			2017		
编号	数量	比例	编号	数量	比例	编号	数量	比例
1	0	0.0%	1	0	0.00%	1	0	0.00%
2	23	0.1%	2	29	0.21%	2	27	0.25%

3	0	0.0%	3	0	0.00%	3	0	0.00%
4	0	0.0%	4	0	0.00%	4	0	0.00%
5	3952	26.%	5	3483	25.6%	5	3145	28.8%
6	8381	56.%	6	7603	55.9%	6	5465	50.1%
7	0	0.0%	7	0	0.00%	7	1	0.01%
8	703	4.7%	8	630	4.64%	8	656	6.02%
9	387	2.5%	9	335	2.46%	9	305	2.80%
10	34	0.2%	10	13	0.10%	10	23	0.21%
11	10	0.07%	11	8	0.06%	11	11	0.10%
12	8	0.05%	12	10	0.07%	12	10	0.09%
13	1466	9.80%	13	1481	10.9%	13	1254	11.5%

### 6.2.2 时空与蔓延特性分析

由于各地区宗教，政治，文化的差异大小不同，世界不同地区的恐怖袭击的分布情况和发生频率有着很大的不同，世界各地区恐怖袭击发生总体分布情况如图 6.5 所示。

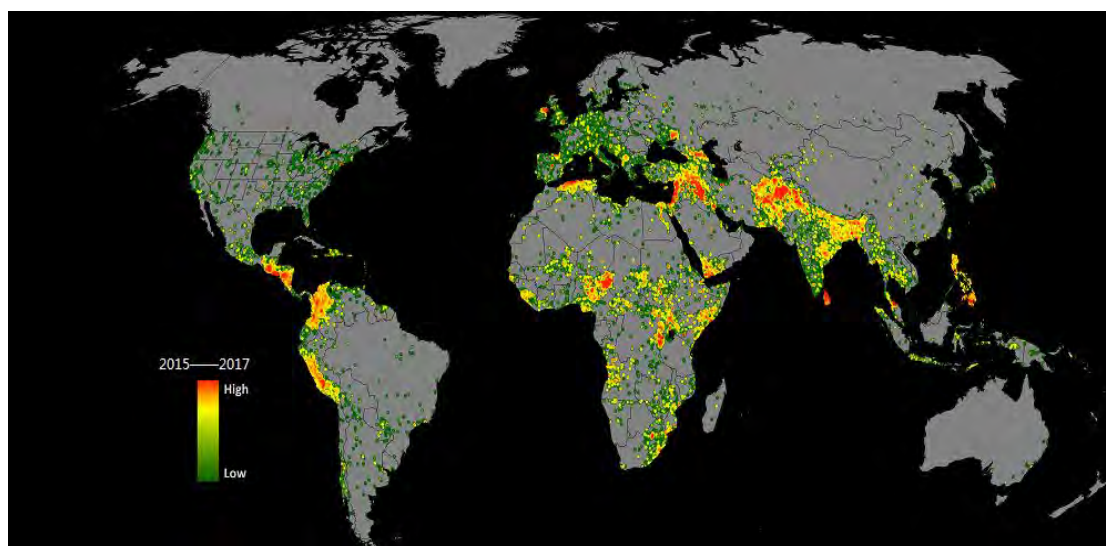


图 6.5 世界各地区恐怖袭击分布图<sup>[13]</sup>

图上颜色越深越密集的地方表明该地区恐怖袭击事件发生越频繁，对 2015 年—2017 年各地区的恐怖事件的统计结果如图 6.6 和图 6.7 所示。

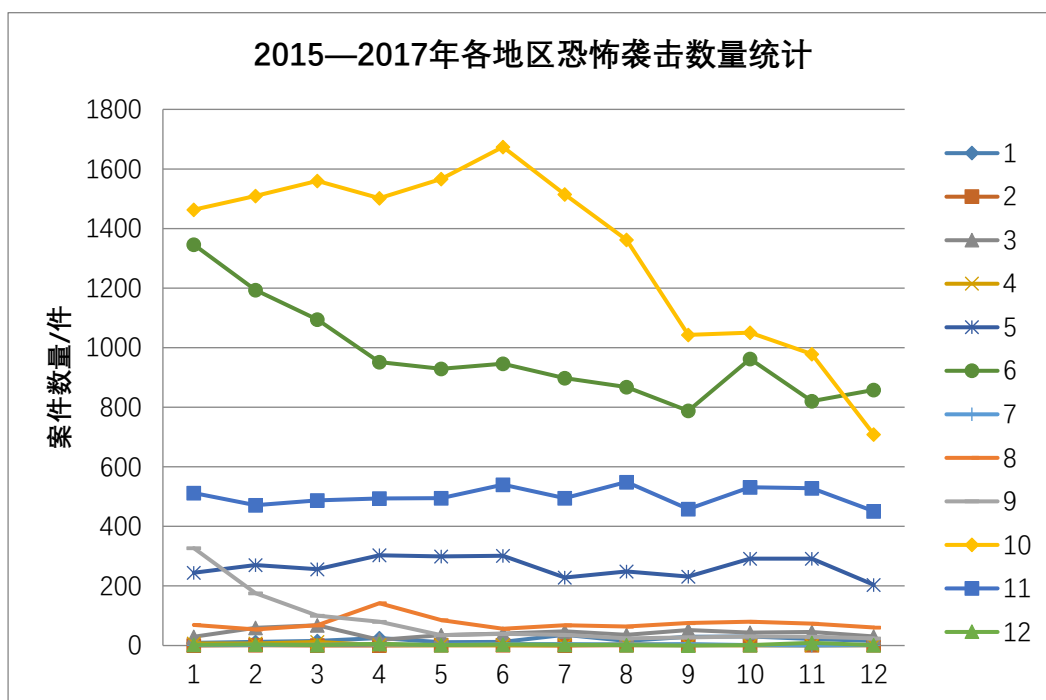


图 6.6 2015 年各地区恐怖袭击数量统计

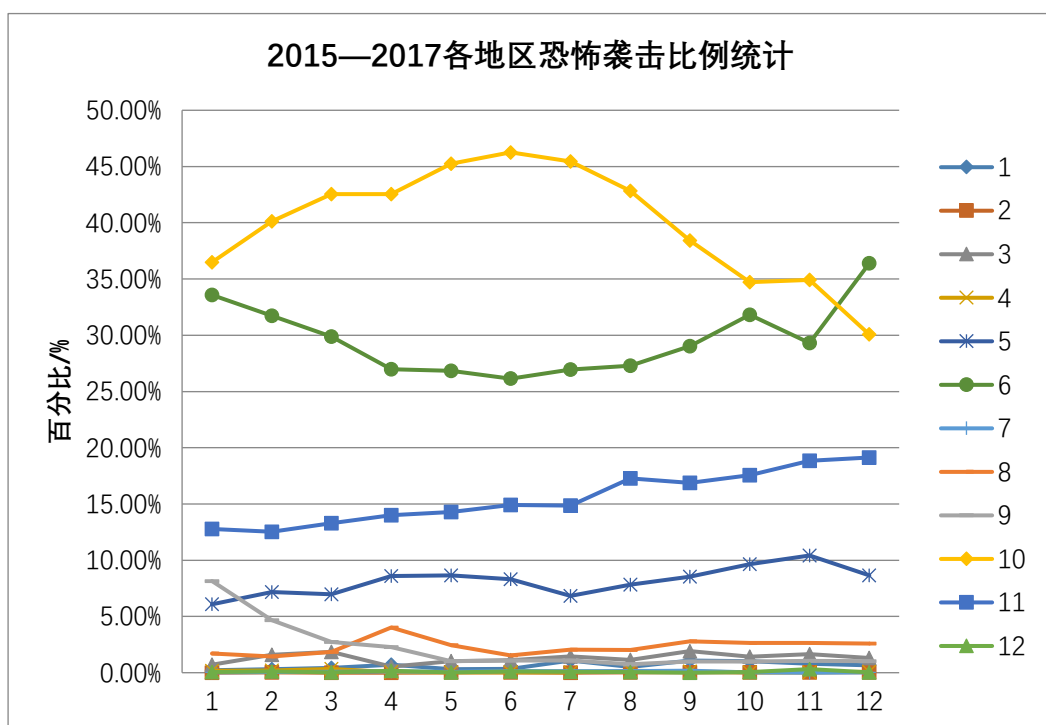


图 6.7 2015 年各地区恐怖袭击数量统计

由图 6.6 和图 6.7 可知，恐怖袭击发生最集中的前两个地区为中东和北非（10）和南亚（6），虽然随着国际恐怖袭击环境的改善，各大地区的恐怖事件总体有所下降，但是南亚（6），撒哈拉以南的非洲（11）以及东南亚（5）未见明显下降趋势，并且其所占比例还有升高的趋势，尤其是南亚地区，这将是下一步反恐的重点。东欧（9）近两年恐怖事件所占的比例有所下滑。

6.3 基于灰色预测的恐怖袭击预测

通过对 2015 年—2018 年 12 个季度的恐怖袭击进行统计，可得各个季度的恐怖袭击事件数量的分布如表 6.3 所示：

表 6.3 各个季度恐怖袭击事件数量分布

季度编号	1	2	3	4	5	6
案件数量/ 千	4.01	3.8744	3.7287	3.5885	3.4535	3.3236
季度编号	7	8	9	10	11	12
案件数量/ 千	3.1986	3.0783	2.9626	2.8511	2.7439	2.6407

为了获得未来恐怖袭击案件数量随时间变化关系，需要建立预测模型对未来进行预测，分别采用数据拟合和灰色预测模型进行建模<sup>[10]</sup>。

数据拟合采用 Matlab 数据拟合工具箱进行，通过尝试不同的拟合函数，发现线性拟合和幂函数拟合比较符合求解要求，其拟合结果如图 6.8 所示：

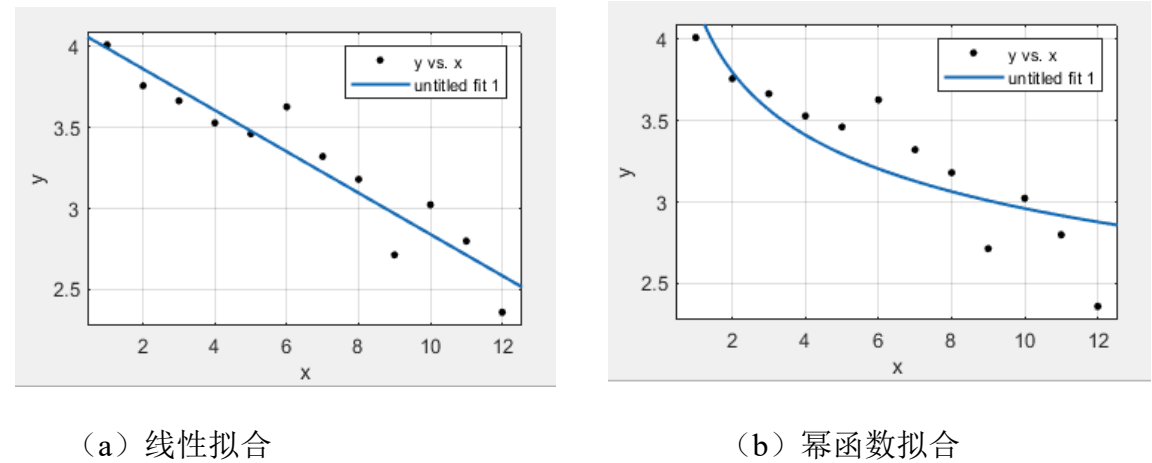


图 6.8 函数拟合曲线

经过运算后函数拟合结果与求解参数如表 6.4 所示：

表 6.4 函数参数求解

函数形式	参数求解 (1)	参数求解 (2)
$f(x) = p_1x + p_2$	-0.1276; 4.117	-0.1574; 3.891
$f(x) = a * x^b$	4.228; -0.1548	3.765; -0.2198

由于恐怖袭击案件的预测带有明显时间序列，且为了数据统计对每个季度的案件进行了合并，一共获得了 12 组数据，属于小数据样本因此考虑使用回测预测模型，灰色预测模型的一般原理和公式在这里不做赘述，预测结果如图 6.9 所示：

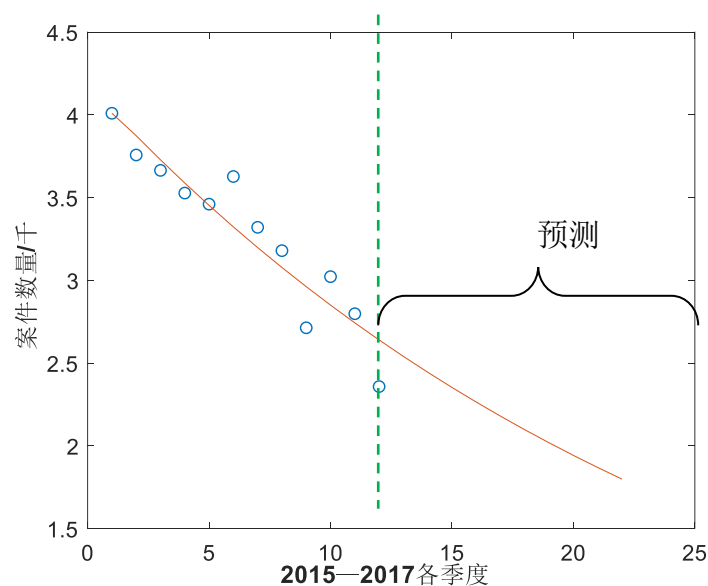


图 6.9 灰色预测模型预测结果

#### 6.4 模型验证

用建立的三个模型，灰色预测，线性拟合，幂函数拟合对未来 10 个季度的恐怖袭击案件数量进行预测分析，其对比结果表 6.5 所示：

表 6.5 不同函数预测结果对比

季度编号	灰色预测/ 千件	线性回归 /千件	幂函数/ 千件
1	4.01	4.01	4.01
2	3.8744	3.8744	3.8744
3	3.7287	3.7287	3.7287
4	3.5885	3.5885	3.5885
5	3.4535	3.4535	3.4535
6	3.3236	3.3236	3.3236
7	3.1986	3.1986	3.1986
8	3.0783	3.0783	3.0783
9	2.9626	2.9626	2.9626
10	2.8511	2.8511	2.8511
11	2.7439	2.7439	2.7439
12	2.6407	2.6407	2.6407
13	2.5414	2.4582	2.8425
14	2.4458	2.3306	2.8101
15	2.3538	2.203	2.7802
16	2.2653	2.0754	2.7526
17	2.1801	1.9478	2.7269
18	2.0981	1.8202	2.7028
19	2.0192	1.6926	2.6803
20	1.9432	1.565	2.6591
21	1.8702	1.4374	2.6391



各函数拟合结果分析如表 6. 6 所示：

表 6. 6 不同函数拟合结果分析

	SSE	R-square	Adjust	RMSE
线性拟合	0. 2722	0. 8954	0. 8849	0. 165
幂函数拟合	0. 7042	0. 7294	0. 7023	0. 2654
灰色预测	0. 2143	0. 9078	0. 9223	0. 158

由表 6. 6 可以看出灰色预测效果最好，相关性强，误差小，线性拟合结果次之，幂函数拟合最差，实际上回灰色预测对于时间序列，小样本预测具有较好的准确性，利用灰色预测模型又对南亚地区、中东与北非以及东欧等变化较大地区的未来恐怖袭击数量进行了预测，预测结果如图 6. 10 所示：

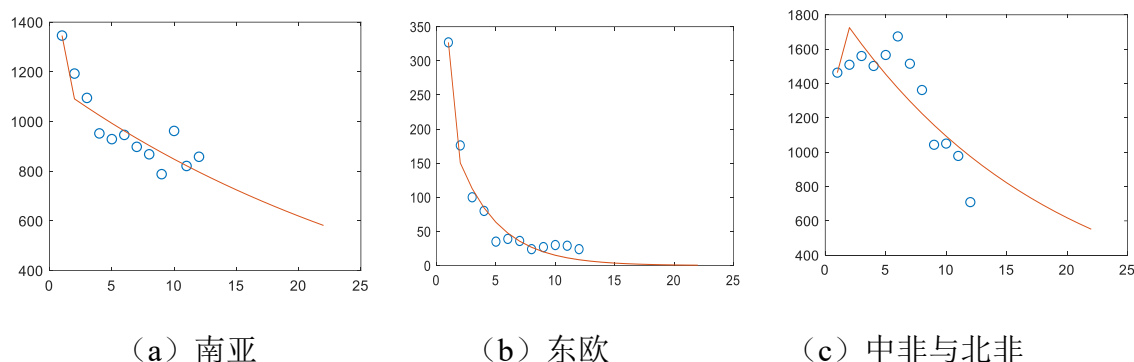


图 6. 10 变化较大地区恐怖袭击案件数量预测

## 6.5 反恐斗争的见解和建议

通过特征可视化统计和灰色预测模型对未来三年恐怖袭击事件发生的原因、时空特性、蔓延特性等进行了深入地研究，得出了如下的结论和建议：

(1) 全球反恐环境得到改善，无论是恐怖袭击的数量、伤亡人数和危害等级都呈现较为明显的下滑趋势。

(2) 中东和北非以及南亚地区仍然是恐怖袭击发生的主要地区，而且这一现状短期将难以改变，近三年中东和北非相较与之前恐怖袭击得到一定的缓和，但是南亚和撒哈拉以南的非洲地区近些年恐怖活动比例却呈现缓慢增长的趋势，这将是下一步反恐的重点观察位置。

(3) 恐怖袭击受害者对象依然是宗教、军事和警察居多，且采取的攻击方式多为轻武器和轰炸、爆炸，其三年以来比例维持相对稳定，表现出较强的关联特征，通过分析攻击对象和攻击方式有助于对同一犯罪集团实施的不同犯罪案件进行归类，有利于提高破案效率。

(4) 恐怖主义进行攻击的方式相对单一，其中轻型武器、爆炸、燃烧武器这三类就约占据了恐怖袭击攻击方式的九成，如果能对上述武器的源头进行遏制，将会有效地遏制恐怖主义的发展。



## 七、问题四的模型建立与求解

### 7.1 问题分析

问题四要求利用数学模型对给出的数据进行进一步挖掘，获得数据中的额外隐含信息。实际上在问题一中已经利用主成分分析算法对恐怖袭击事件的分级特征进行了降维和特征提取，并利用聚类算法对不同类型的恐怖袭击事件进行了定级，建立了衡量恐怖袭击危害的特征向量与分级的数据对应关系，每一类特征向量对应着一个危害等级标签，这将原先的无监督聚类问题，转化成了有监督的分类问题。通过机器学习模型对数据集进行训练，以此来验证基于问题一的危害分级的准确性和，若经过训练获得较好的分类准确率，则说明问题一中的分类方法是正确合理的，反之不合理。同时利用近邻分析法（NCA）可以求解各类特征向量对于分类和聚类的贡献权重，并与问题一中贡献率进行对比，进一步进行数据分析和降维处理。

### 7.2 模型建立

考虑到用于恐怖等级的训练的数据样本是离散的，且经过主成分分析完成了特征提取和降维处理，因此考虑采用机器学习的算法进行网络训练，其一般的网络如图 7.1 所示。其中恐怖等级特征向量作为网络的输入，等级标签作为网络的输出，输入信号从输入层经隐含层逐层处理，直至输出层，每一层神经元只影响下一层神经元状态。如果输出得不到期望输出，则转入反向传播<sup>[11]</sup>。

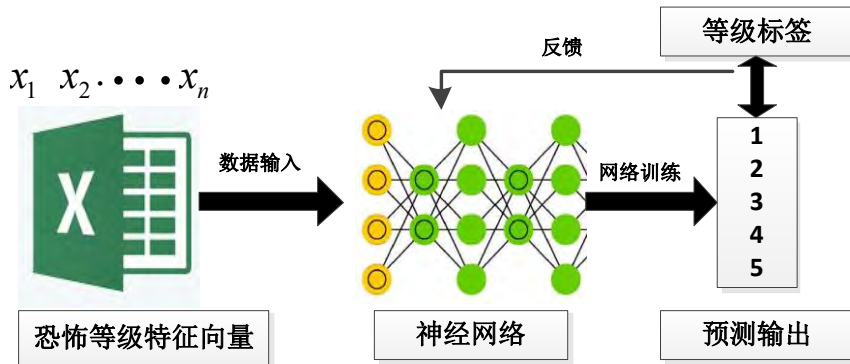


图 7.1 机器学习网络结构图

步骤 1: 网络初始化。根据系统输入输出序列  $(X, Y)$  确定网络输入层节点数  $n$ 、隐含层节点数  $l$ ，输出层节点数  $m$ ，初始化输入层、隐含层和输出层神经元之间的连接权值  $w_{ij}$ ， $w_{jk}$  初始化隐含层阈值  $a$ ，输出层阈值  $b$ ，给定学习速率和神经元激励函数。

步骤 2: 隐含层输出计算。根据输入变量  $X$ ，输入层和隐含层连接权值  $w_{ij}$  以及隐含层阈值  $a$ ，计算隐含层输出  $H$ 。

$$H_j = f\left(\sum_{i=1}^n \omega_{ij} x_i - a_j\right) \quad j=1, 2, \dots, l \quad (7.1)$$

式中， $l$  为隐含层节点数； $f$  为隐含层激励函数，该函数有多种表达形式，其

中默认的激励函数为：

$$f(x) = \frac{1}{1 + e^{-x}} \quad (7.2)$$

步骤 3：输出层计算。根据隐含层输出  $H$ ，连接权值  $w_{jk}$  和阈值  $b$ ，计算 BP 神经网络预测  $O_k$ 。

$$O_k = \sum_{j=1}^l H_j w_{jk} - b_k \quad k = 1, 2, \dots, m \quad (7.3)$$

步骤 4：误差计算。根据网络预测输出  $O$  和期望输出  $Y$ ，计算网络预测误差  $e$ 。

$$e_k = Y_k - O_k \quad k = 1, 2, \dots, m, \quad (7.4)$$

步骤 5：权值更新。根据网络预测误差  $e$  更新网络连接权值  $w_{ij}$ ， $w_{jk}$ 。

$$\omega_{ij} = \omega_{ij} + \eta H_j (1 - H_j) x(i) \sum_{k=1}^m \omega_{jk} e_k \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, l \quad (7.5)$$

$$\omega_{jk} = \omega_{jk} + \eta H_j e_k \quad j = 1, 2, \dots, l \quad k = 1, 2, \dots, m \quad (7.6)$$

式中， $\eta$  为学习速率。

步骤 6：阈值更新。根据网络预测误差  $e$  更新网络节点阈值  $a$ ， $b$ 。

$$a_j = a_j + \eta H_j (1 - H_j) \sum_{k=1}^m \omega_{jk} e_k \quad (7.7)$$

$$b_k = b_k + e_k \quad k = 1, 2, \dots, m, \quad (7.8)$$

步骤 7：判断算法迭代是否结束，若没有结束，返回步骤 2。

### 7.3 网络训练与测试

为了充分验证各类机器学习算法对分类准确率，本文采用 Matlab2018 自带的 Classification Learner 工具箱对数据进行训练，该工具箱包集合了包括各类 SVM，决策树，BP 网络等在内的一共 22 类分类器，如图 7.2 所示，为了缩减网络训练的时间，整个训练过程采用并行加速方式进行训练<sup>[12]</sup>。

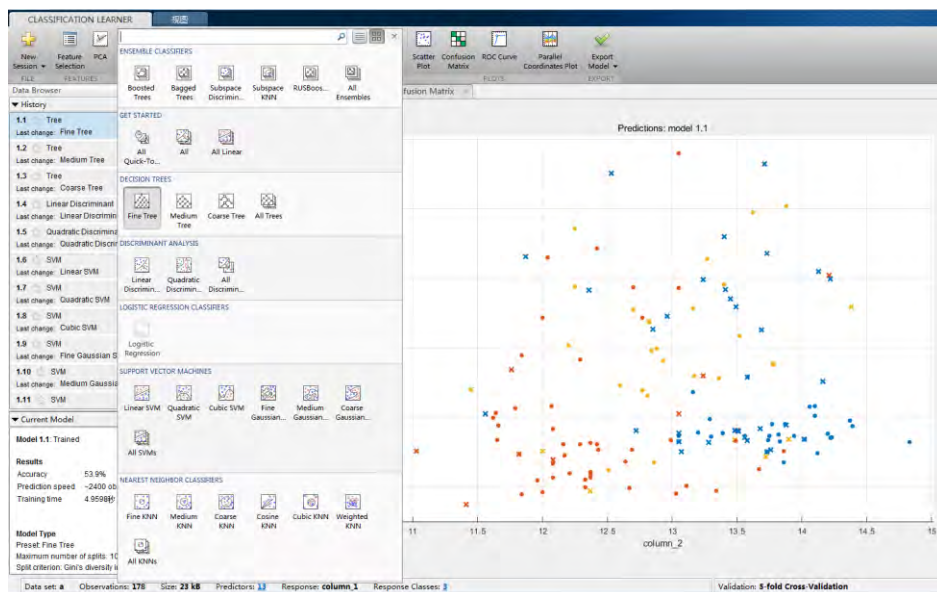


图 7.2 Matlab 机器学习工具箱

由第一问可知不同等级的恐怖袭击案件的聚类结果如图 7.3 所示，很明显如果随机抽取 70%的数据作为训练数据，30 的数据作为测试数据，由于各个分类级别案件数量的差异巨大，很可能导致某一类案件的无法获得相应的训练数据（如等级 1 和等级 2）。为了充分发挥各类案件的数据信息，对训练样本和测试样本的划分如表 7.1 所示。各分类器的训练结果如表 7.2 所示

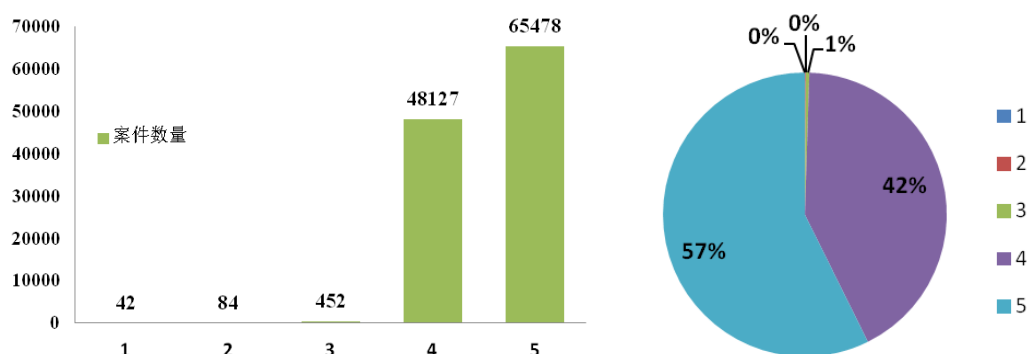


图 7.3 聚类结果图

表 7.1 样本划分

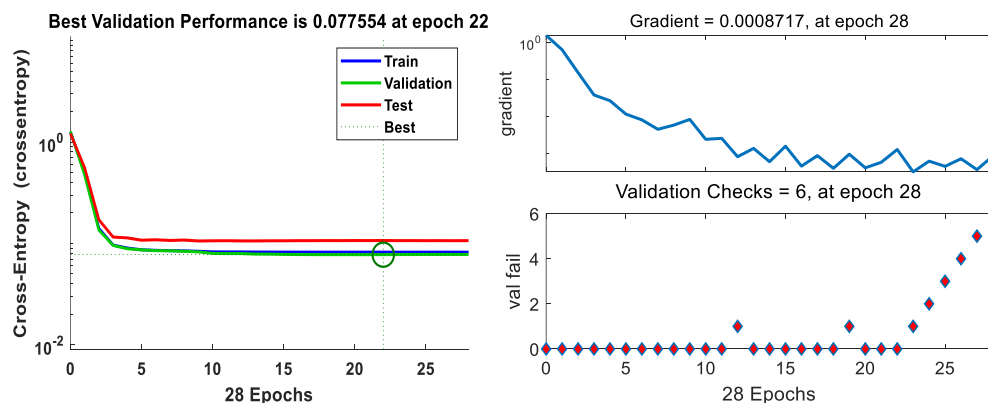
	1 级	2 级	3 级	4 级	5 级
样本总量	42	84	452	48127	65478
训练样本	42	84	452	33689	45835
比例	100%	100%	100%	70%	70%
选取方式	全部	全部	全部	随机	随机
测试样本	12	24	136	14438	19643
比例	30%	30%	30%	30%	30%
选取方式	随机	随机	随机	随机	随机

表 7.2 不同机器学习模型分类性能对比

编号	模型	运行时间 /s	分类正确 率/%
----	----	------------	-------------

1	Fine tree	20.36	88.7
2	Medium tree	20.09	88.2
3	Coarse tree	12.93	83.6
4	Linear Discriminant	16.82	75.2
5	Quadratic Discriminant	27.71	74.8
6	Linear SVM	527.32	88.2
7	Quadratic SVM	700.15	89.4
8	Cubic SVM	507.79	87.1
9	Fine Gaussian SVM	499.42	88.6
10	Medium Gaussian SVM	524.58	87.8
11	Coarse Gaussian SVM	800.72	85.5
12	Fine KNN	17.62	74.5
13	Medium KNN	1.34	74.5
14	Coarse KNN	11.26	88.6
15	Cosine KNN	17.08	74.5
16	Cubic KNN	14.52	74.5
17	Weighted KNN	15.37	74.5
18	Boosted KNN	17.21	88.5
19	Bagged Trees	36.45	88.6
20	Subspace Discriminant	38.55	75.4
21	Subspace KNN	17.59	74.1
22	RUSBoosted Tress	28.72	88.5

由表 7.2 可知不同机器学习对恐怖袭击危害等级识别的准确率 74.1%—89.4%，其中基于 SVM 的分类器平均识别准确率 87.76%，识别精度较高，但平均训练时间为 9.8min，训练时间偏长；基于决策树的分类算法平均识别准确率为 86.4%，稍低于 SVM，但是训练时间较短，平均时间为 17.7s；而 KNN 算法相较于其他算法平均识别准确率较低。部分训练与测试过程如图 7.4 所示：



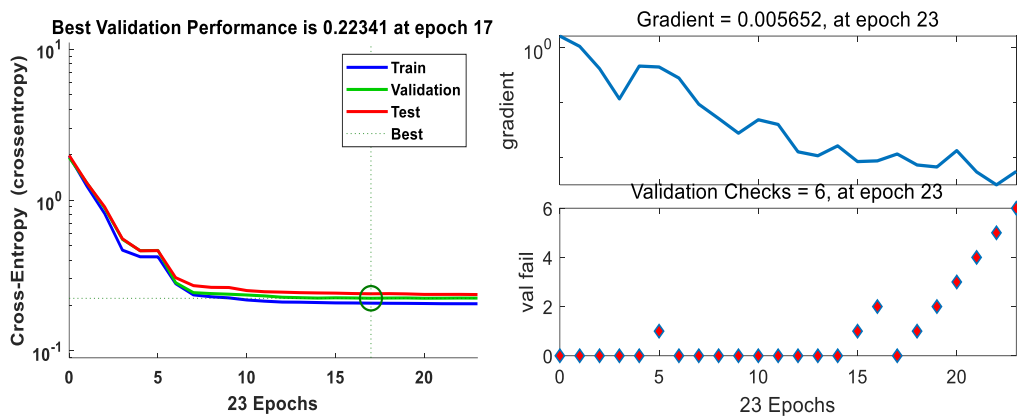


图 7.4 部分模型训练与测试结果

为了验证机器学习训练结果识别的准确性，用剩余的 30%的训练数据对获得的机器学习模型进行测试部分模型试验测试结果如图 7.5 所示：

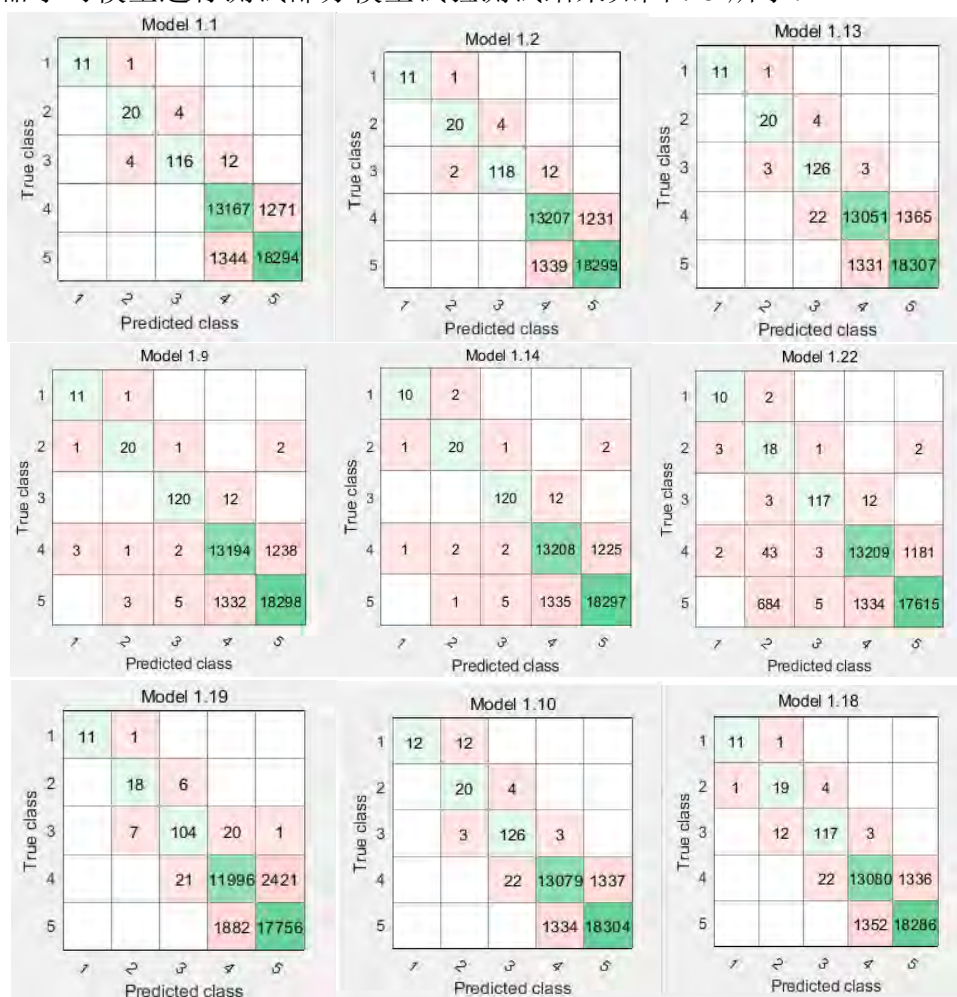


图 7.5 部分模型试验测试结果图

其中测试评价矩阵的横坐标代表的是机器学习神经网络预测的恐怖袭击事件的等级（1—5 级），纵坐标代表的是训练数据中该恐怖袭击事件的等级标签（1—5），对角线蓝色元素代表的是当预测等级与标签等级一直时，则表示分类正确，颜色的深浅代表该类等级测试结果的数据量，数据量越多颜色越深，反之颜色越浅。红色代表分类错误的情况的集合，颜色越深的部分代表分类错误的数据越大。以 Model 1 模型的为例，第 2 行，第 3 列的“4”代表的是有 4 类恐怖

袭击事件样本等级标签为 2 级，但是在实际模型分类时被划分为第 3 级。其他的情况以此类推。

通过测试结果评价矩阵矩阵可以看出通过网络训练获得的恐怖袭击等级分类模型就有较好的准确性，其等级分类结果如表 7.3 所示

表 7.3 等级分类结果图

模型编号	1 级		2 级		3 级	
	正确	错误	正确	错误	正确	错误
1	11	1	20	4	116	16
2	11	1	20	4	118	14
9	11	1	20	4	120	12
10	12	0	20	4	126	6
13	11	1	20	4	126	6
14	10	2	20	4	120	12
18	11	1	19	5	117	15
19	11	1	18	6	104	28
22	10	2	18	6	117	15

模型编号	4 级		5 级		总准确率
	正确	错误	正确	错误	
1	13167	1271	18294	1344	86.8%
2	13207	1231	18299	1339	87.0%
9	13194	1244	18298	1440	84.2%
10	13079	1359	18304	1334	87.4%
13	13051	1387	18037	1331	87.8%
14	13208	1230	18297	1341	86.9%
18	13079	1359	18286	1352	84.6%
19	11996	2442	17756	1882	83.1%
22	13209	1229	18286	1352	85.1%

综上所述基于机器学习的恐怖等级识别算法准确率较高，这也很好的验证了第一问危险等级分类方法的合理性。

### 7.3.1 NCA 近邻分析

NCA 是一种基于邻域分量的特征选择方法，在有监督的机器学习方法中，通过 NCA 多步迭代算法对用于分类的特征向量进行分析，以此获得不同的特征向量的分类中的权重大小，可根据权重的大小对原始的特征向量进行进一步筛选，进一步对数据进行降维，对整个训练网络进行优化。

在第一问中我们已经通过 PCA 降维算法对影响恐怖袭击的危害等级的特征向量进行了排序和权重计算，在第四问中我们再利用 NCA 对影响恐怖袭击等级的特征向量的排序和权重进行进一步的讨论，以此来验证第一问中的计算结果的合理性和有效性。

### 7.3.2 NCA 计算参数设置

Matlab 2018 有用于 NCA 近邻分析的工具箱，函数命令为 fscnca(x,y),其中 x

代表特征向量， $y$  代表样本标签（危害等级），因为  $nca$  是一中迭代算法，在进行运算时需要对一些参数进行设置，包括迭代方法，最小批量处理，得跌周期等。

考虑到样本的数量和计算精度的要求，对 NCA 计算参数的设置如下，为提高计算精度网络的迭代方式采用 SGD 随机梯度下降的计算方式，最小计算批量（MiniBatchSize）为 50，PassLimitd 等于 10，调谐子集合为 150，迭代周期为 20。网络运行迭代过程中 loss 不断减小，直到稳定达到误差要求。其迭代过程如图 7.6 所示，计算得出的 NCA 与 PCA 特征向量权重/得分如图 7.7 所示，最终各类结果 PCA 权重及排名、NCA 打分与排名结果如表 7.4 所示。

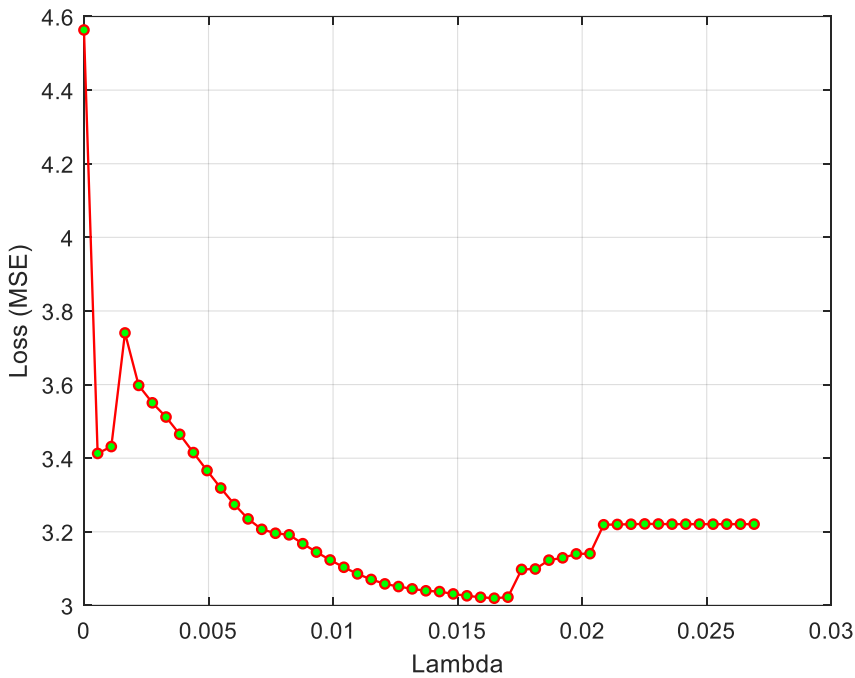


图 7.6 NCA 计算迭代曲线

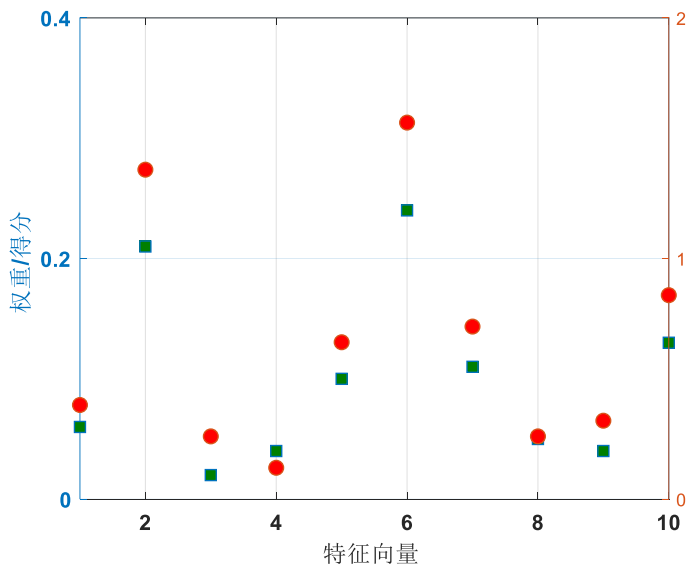


图 7.7 NCA 与 PCA 特征向量权重/得分



表 7.4 打分与排名结果

编号	含义	PCA 权重	排名	NCA 打分	排名
1	武器种类	0.06	6	0.3912	6
2	地区代码	0.21	2	1.3692	2
3	成功攻击	0.02	10	0.2608	8
4	攻击类型	0.04	8	0.1304	10
5	受害者类型	0.13	3	0.8476	3
6	死亡总数	0.24	1	1.5648	1
7	受伤总数	0.11	4	0.7172	4
8	凶手死亡人数	0.05	7	0.2608	9
9	凶手受伤总数	0.04	9	0.326	7
10	财产损失等级	0.1	5	0.652	5

由图 7.7 和表 7.4 可知, PCA 权重和 NCA 打分总体趋势是一致的, 只是略有不同, PCA 认为凶手死亡人数>攻击类型>凶手受伤人数>成功攻击, NCA 则认为凶手受伤人数>成功攻击>凶手死亡人数>攻击类型。

## 八、参考文献

- [1]陈伟,杨锐,何涛,王朔,陈江萍.大数据环境下科技情报研究的新模式[J].科技导报,2018,36(16):78-85.
- [2]龚芳海,李文彪.基于互联网的大数据挖掘关键技术分析[J].无线互联科技,2018,15(04):59-60.
- [3]张磊.基于聚类分析的陆军装备质量等级划分[J].科技经济导刊,2018,26(04):117.
- [4]李勇男.基于雅卡尔系数的反恐情报聚类分析[J].现代情报,2018,38(01):51-55.
- [5]吴绍忠.基于聚类分析的反恐情报中潜在恐怖团伙发现技术[J].警察技术,2016(06):18-21.
- [6]张玉超.公安重点关注对象的聚类分析研究[D].山东大学,2015.
- [7]吴笛,杜云艳,易嘉伟,魏海涛,莫洋.基于密度的轨迹时空聚类分析[J].地球信息科学学报,2015,17(10):1162-1171.
- [8]郭文月.基于全球恐怖主义数据库的社会安全事件时空关联分析方法研究[D].解放军信息工程大学,2015.
- [9]魏海亮,王振华.我国安全形势与反恐情报战略构建——基于国际恐怖主义的视角[J].情报杂志,2015,34(04):13-18.
- [10]黄金川,陈守强.中国城市群等级类型综合划分[J].地理科学进展,2015,34(03):290-301.
- [11]付举磊.基于开源情报的恐怖活动及反恐策略研究[D].国防科学技术大学,2014.
- [12]靳高风.2013 年中国犯罪形势分析及 2014 年预测[J].中国人民公安大学学报(社会科学版),2014,30(02):8-15.
- [13]王策.一种基于 k-means 算法和关联规则的缺失数据填补方法[D].哈尔滨工程大学,2014.



- [14]鲁峰,黄金泉.基于灰色关联聚类的特征提取算法[J].系统工程理论与实践,2012,32(04):872-876.
- [15]罗艳. 基于 DEA 方法的指标选取和环境效率评价研究[D].中国科学技术大学,2012.
- [16]王贤. 基于逻辑回归的案件关联分析[D].西南大学,2009.
- [17]陈治国,张春元.基于聚类分析的学生等级制成绩评定方法[J].电脑知识与技术,2006(02):131-132+137.
- [18]丁天赞. 突发公共卫生事件应急反应体系和运行机制的研究[D].山东大学,2005.

## 附录

K-means 代码
<pre> #include &lt;stdio.h&gt; #include &lt;string.h&gt; #include &lt;stdlib.h&gt; #include &lt;math.h&gt; #include &lt;time.h&gt;  #define N 11 #define K 3  typedef struct {     float x;     float y; }Point;  int center[N];  /// 判断每个点属于哪个簇  Point point[N] = {     {2.0, 10.0},     {2.0, 5.0},     {8.0, 4.0},     {5.0, 8.0},     {7.0, 5.0},     {6.0, 4.0},     {1.0, 2.0},     {4.0, 9.0},     {7.0, 3.0}, </pre>

```

        {1.0, 3.0},
        {3.0, 9.0}
    };

    Point mean[K];    /// 保存每个簇的中心点

    float getDistance(Point point1, Point point2)
    {
        float d;
        d = sqrt((point1.x - point2.x) * (point1.x - point2.x) + (point1.y - point2.y)
* (point1.y - point2.y));
        return d;
    }

    /// 计算每个簇的中心点
    void getMean(int center[N])
    {
        Point tep;
        int i, j, count = 0;
        for(i = 0; i < K; ++i)
        {
            count = 0;
            tep.x = 0.0;    /// 每算出一个簇的中心点值后清 0
            tep.y = 0.0;
            for(j = 0; j < N; ++j)
            {
                if(i == center[j])
                {
                    count++;
                    tep.x += point[j].x;
                    tep.y += point[j].y;
                }
            }
            tep.x /= count;
            tep.y /= count;
            mean[i] = tep;
        }
        for(i = 0; i < K; ++i)
        {
            printf("The new center point of %d is : \t( %f, %f )\n", i+1, mean[i].x,
mean[i].y);
        }
    }
}

```

```

/// 计算平方误差函数
float getE()
{
    int i, j;
    float cnt = 0.0, sum = 0.0;
    for(i = 0; i < K; ++i)
    {
        for(j = 0; j < N; ++j)
        {
            if(i == center[j])
            {
                cnt = (point[j].x - mean[i].x) * (point[j].x - mean[i].x) +
(point[j].y - mean[i].y) * (point[j].y - mean[i].y);
                sum += cnt;
            }
        }
    }
    return sum;
}

/// 把 N 个点聚类
void cluster()
{
    int i, j, q;
    float min;
    float distance[N][K];
    for(i = 0; i < N; ++i)
    {
        min = 999999.0;
        for(j = 0; j < K; ++j)
        {
            distance[i][j] = getDistance(point[i], mean[j]);

            /// printf("%f\n", distance[i][j]); // 可以用来测试对于每个点
与 3 个中心点之间的距离
        }
        for(q = 0; q < K; ++q)
        {
            if(distance[i][q] < min)
            {
                min = distance[i][q];
                center[i] = q;
            }
        }
    }
}

```

```

        printf("( %.0f, %.0f )\t in cluster-%d\n", point[i].x, point[i].y, center[i]
+ 1);
    }
    printf("-----\n");
}

int main()
{
    int i, j, n = 0;
    float temp1;
    float temp2, t;
    printf("-----Data sets-----\n");
    for(i = 0; i < N; ++i)
    {
        printf("\t( %.0f, %.0f )\n", point[i].x, point[i].y);
    }
    printf("-----\n");

    /*
    可以选择当前时间为随机数
    srand((unsigned int)time(NULL));
    for(i = 0; i < K; ++i)
    {
        j = rand() % K;
        mean[i].x = point[j].x;
        mean[i].y = point[j].y;
    }
    */

    mean[0].x = point[0].x;    /// 初始化 k 个中心点
    mean[0].y = point[0].y;

    mean[1].x = point[3].x;
    mean[1].y = point[3].y;

    mean[2].x = point[6].x;
    mean[2].y = point[6].y;

    cluster();    /// 第一次根据预设的 k 个点进行聚类
    temp1 = getE();    /// 第一次平方误差
    n++;    /// n 计算形成最终的簇用了多少次

    printf("The E1 is: %f\n\n", temp1);

    getMean(center);

```

```

        cluster();
        temp2 = getE();          /// 根据簇形成新的中心点, 并计算出平方误差
        n++;

        printf("The E2 is: %f\n", temp2);

        while(fabs(temp2 - temp1) != 0)    /// 比较两次平方误差 判断是否相等, 不相等继续迭代
        {
            temp1 = temp2;
            getMean(center);
            cluster();
            temp2 = getE();
            n++;
            printf("The E%d is: %f\n", n, temp2);
        }

        printf("The total number of cluster is: %d\n", n);    /// 统计出迭代次数
        system("pause");
        return 0;
    }

```

#### PCA 主成分分析代码

```

function main()
%*****主成份分析*****
%see also http://www.matlabsky.com
%读入文件数据
X=load('data.txt');
%标准化处理
[p,n]=size(X);
for j=1:n
    mju(j)=mean(X(:,j));
    sigma(j)=sqrt(cov(X(:,j)));
end
for i=1:p
    for j=1:n
        Y(i,j)=(X(i,j)-mju(j))/sigma(j);
    end
end
sigmaY=cov(Y);
%求 X 标准化的协差矩阵的特征根和特征向量
[T,lambda]=eig(sigmaY);
disp('特征根(由小到大):');
disp(lambda);

```

```

disp('特征向量:');
disp(T);
%方差贡献率;累计方差贡献率
Xsum=sum(sum(lambda,2),1);
for i=1:n
    fai(i)=lambda(i,i)/Xsum;
end
for i=1:n
    psai(i)= sum(sum(lambda(1:i,1:i),2),1)/Xsum;
end
disp('方差贡献率:');
disp(fai);
disp('累计方差贡献率:');
disp(psai);

```

#### 逻辑回归代码

```

#include "main.h"
int main()
{
    char                                *file                                =
"C:\\Users\\Administrator\\Desktop\\machine_learning\\wpbc.data";
    DataSample *data = new DataSample[sampleNum];
    double *logisW = new double[attriNum+1];

    if( -1!=ReadData( data,file ) )
    {
        Logistic( data,logisW );
    }

    for(int i=0;i<(attriNum+1);++i)
    {
        printf("%f\t",logisW[i]);
    }
    printf("\n\n");

    int correct = 0;
    int sum = 0;
    for(int i=trainNum;i<sampleNum; ++i)
    {
        ++sum;
        bool eva = Predict(data[i],logisW);
        if(eva)
            ++correct;
    }
}

```

```

    double rp = double(correct)/sum;
    printf("the right correction: %f\n",rp);

    delete []data;
    delete []logisW;

    return 0;
}

#ifdef MAIN_H
#define MAIN_H

#include "stdio.h"
#include "stdlib.h"
#include "iostream"
#include "string"
#include "string.h"
#include <sstream>
#include <memory.h>

#include "math.h"

using namespace std;

#define maxClassLabelNum 10;
int curLabelNum = 0;

const double alph = 0.3; //set the newton gradient algorithm fixed step
const int attriNum = 33;
const int sampleNum = 198;
int trainNum = 140;

struct DataSample
{
    double attriValue[attriNum];
    bool classLabel;
};

double StringTodouble(char * src)
{
    double a;
    stringstream str;

```

```

        str<<src;
        str>>a;
        str.clear();
        return a;
    }

int ReadData( DataSample* data, char *file)
{
    FILE *pFile;
    char buf[1024];
    pFile = fopen(file,"rt");
    if(pFile==NULL)
    {
        printf("the data file is not existing:  %s\n", file);
        return -1;
    }

    int row = 0;    //data line
    int cloumn = 0; //data attribute
    char delim[] = ","; //data delimiter
    char *tmpdata = NULL; //data cache

    while(!feof(pFile)&&row<sampleNum)
    {
        buf[0] = '\0';
        fgets(buf,1024,pFile);

        if( buf[strlen(buf)-1]=='\n' )
        {
            buf[strlen(buf)-1]='\0';
        }

        //the first column is non-used,and second column is class label;
        for( int column = 0;column<(attriNum+2);++column )
        {
            if( column==0 )
            {
                tmpdata = strtok(buf,delim);
                continue;
            }
            else if( column==1 )
            {

```



```

        tmpdata = strtok(NULL,delim);

        if( tmpdata[0]=='R' )
            data[row].classLabel = 1; //R:1;  N:0
        else
            data[row].classLabel = 0;

    }
    else
    {
        tmpdata = strtok(NULL,delim);

        if(tmpdata[0]!='?')// '?' mean the loss attribute value
            data[row].attriValue[column-2] =
StringTodouble(tmpdata);
        else
            data[row].attriValue[column-2] = -1000;
    }
}
++row;

}

return 1;
}

void Normalize( DataSample* data )
{
    double attriMinValue[attriNum];
    double attriMaxValue[attriNum]; //for normalization (x-xmin)/(xmax-xmin)

    //think about the first sample is none-loss
    //get the min and max value of each attribute without thinking about the loss
attribute
    for( int i=0;i<attriNum;++i )
    {
        attriMinValue[i] = data[0].attriValue[i];
        attriMaxValue[i] = data[0].attriValue[i];
    }

    for( int row = 1; row < sampleNum; ++row )
        for( int column = 0; column < attriNum; ++column )
        {

```

```

        if( data[row].attriValue[column] > attriMaxValue[column] &&
(data[row].attriValue[column]+1000)>0.0001 )
            attriMaxValue[column] = data[row].attriValue[column];

        if( data[row].attriValue[column] < attriMinValue[column] &&
(data[row].attriValue[column]+1000)>0.0001 )
            attriMinValue[column] = data[row].attriValue[column];
    }

    for( int row = 1; row < sampleNum; ++row )
        for( int column = 0; column < attriNum; ++column )
        {
            if( (data[row].attriValue[column]+1000)>0.0001)
                data[row].attriValue[column] =
(data[row].attriValue[column]-attriMinValue[column])/(attriMaxValue[column]-
attriMinValue[column]);
            else
                data[row].attriValue[column] = 0;//set loss value 0;
        }
    }

void Logistic( DataSample* data, double *logisW )
{
    //memset( logisW,1.0,(attriNum+1)*sizeof(double) );//initial

    for( int i=0;i<(attriNum+1);++i )
    {
        logisW[i] = 1.0;
    }

    Normalize( data );

    double h = 0.0;
    double error = 0.0;
    for( int row=0; row<trainNum; ++row )
    {
        h = 0.0;
        for( int column=0; column<attriNum; ++column )
        {
            h += data[row].attriValue[column]*logisW[column];
        }
        h += logisW[attriNum]*1;
    }
}

```

```

        h = 1/(1+exp(-h));

        error = data[row].classLabel-h;

        for( int column=0; column<attriNum; ++column )
        {
            logisW[column] += error*alph*data[row].attriValue[column];
        }
        logisW[attriNum] = error*alph*1;
    }
}

bool Predict( DataSample sample, double *logisW )
{
    double h = 0.0;
    bool label = 0;
    for( int column=0; column<attriNum; ++column )
    {
        h += sample.attriValue[column]*logisW[column];
    }
    h += logisW[attriNum];

    if( h>0.5 )
        label = 1;
    else
        label = 0;

    if( label==sample.classLabel )
        return 1;
    else
        return 0;
}

```