**基于XXX的XXX方案**

**摘要**

第一段：

首先对标题的内涵进行延伸，然后对论文一句话总括。

中间段：

分问题进行介绍，顺序依次为问题简介🡪方法🡪模型🡪算法🡪结果，具体按照下述模板：

对于XXX问题，利用XXX方法（或理论），建立了XXX模型，设计了XXX算法，得到了XXX结果。

最后一段：

可以对论文的其他内容进行介绍，也可以对论文所做的深入分析进行介绍。当然也可以使用一个有特色的结尾，将论文的模型和算法的性能之类的数据进行总结。

**关键词：** XXX XXX XXX XXX XXX（3-5个）

## 1问题重述

### 1**.1问题背景**

恐怖活动是指[恐怖份子](https://zh.wikipedia.org/wiki/%E6%81%90%E6%80%96%E4%BB%BD%E5%AD%90" \o "恐怖份子)制造的危害社会稳定、危及[平民](https://zh.wikipedia.org/wiki/%E5%B9%B3%E6%B0%91" \o "平民)的生命与财产安全的一切形式的活动，通常表现为针对平民的[大规模伤害](https://zh.wikipedia.org/w/index.php?title=%E5%A4%A7%E8%A6%8F%E6%A8%A1%E5%82%B7%E5%AE%B3&action=edit&redlink=1" \o "大规模伤害（页面不存在）)、[袭击](https://zh.wikipedia.org/wiki/%E8%A2%AD%E5%87%BB" \o "袭击)公共交通工具和[绑架](https://zh.wikipedia.org/wiki/%E7%B6%81%E6%9E%B6" \o "绑架)等形式，与恐怖活动相关的事件通常称为“恐怖事件”、“恐怖袭击”等。是一种较极端的[非对称作战](https://zh.wikipedia.org/wiki/%E9%9D%9E%E5%B0%8D%E7%A8%B1%E4%BD%9C%E6%88%B0" \o "非对称作战)方式。恐怖活动主要以制造大量伤亡、破坏以及动荡社会为主要目的，以宗教或政治为主要诉求。

恐怖主义是人类共同的威胁，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。

### 1.2待解决的问题

问题一：利用全球恐怖主义数据库（GTD）提供的信息，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。利用该模型预测近二十年全球十大恐怖袭击事件及给定恐怖袭击事件的等级。

问题二：通过数学建模方法，对2015、2016年发生的、尚未有组织或个人宣称负责的恐怖袭击事件进行分析，预测这些恐怖袭击事件的制造者。即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，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。再对给定的恐袭事件按嫌疑程度对5个嫌疑人排序。

问题三：建立数学模型，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。

## 2模型假设

针对本问题，建立如下合理假设：

1题目所给数据准确可靠；

2能在主流中文搜索引擎中检索的到的恐怖袭击事件均为较有影响力的恐怖袭击事件

3假设恐怖袭击事件带来的后果不是恐怖袭击危害程度的唯一评价指标。

## 3符号系统

本文中所用到的各符号具体定义如下表：

表4-1 符号系统

|  |  |
| --- | --- |
| 符号 | 定义 |
| A | 判断矩阵 |
| W | 权值向量 |
| Y | 恐怖袭击事件危害性评估值 |
| Cr | 矩阵A 的随机一致性比率 |
| Ci | 一致性指标 |
| Ri | 平均随机一致性指标，它只与矩阵A的阶数 n 有关 |

## 4全局数据清洗

数据清洗包括初步数据清洗，任务一数据清洗和任务二数据清洗。

全局数据清洗旨在将原始数据进行全局清洗，进而得到可适用于任务一、任务二、任务三和任务四的数据。其主要目的是消除数据中的错误、冗余和噪声，确保数据的完整性和标准数值化，从而有利于下一步的数据分析。

全局数据清洗的步骤包括数据剔除，数据完整化，数据数值化，数据标准化。这些步骤的详细意义和操作如下文所述。

### 4.1数据剔除

数据剔除的目的是将原始数据中与所有任务都不相关的错误与冗余数据全部去除。分为两个方向：

1. 纵向特征剔除。

剔除与所有任务都不相关的纵向特征列。具体包含了表示数据来源的特征项（scite1, scite2,scite3,dbsource），与数值编码重复了的描述性语句的特征项（如targtype1\_txt 与targtype1意义重复了，且targtype1已经数值化了，所以targtype1\_txt可以删除），其他的一些不方便使用的描述性语句（如location, summary等）。具体见表1.

表1.

1. 横向样本剔除。

将基本上认为该事件不是恐怖主义行为的样本删除。即如果某事件的疑似恐怖主义（doubtterr）标记为‘0’，则表示基本上不怀疑该事件是恐怖主义行为，所以该样本属于错误样本，可以直接删除，这样更加有利于下一步的数据分析。

### 4.2数据完整化

所有特征项中存在数据缺失的，统一标记为‘unknown’。

### 4.3数据数值化

由于在后续数据分析中主要采用的是xgboost算法，而且本案例中的非数值数据是离散的，没有数值大小意义的，所以这里的数值化编码可以采用二进制编码格式。具体做法为：

对于非数值数据，如一个单词或一段描述性语句的字符串，先统计该特征项总共有多少个分布值f\_num, 然后一一对应地编号至Num = [0…f\_num], 再按照二进制编码格式将Num编码转化为Binary\_Num。

### 4.4数据标准化

对于‘unknown’项，统一转化为数值‘-99’。

对于分布不均匀的数值特征，先统计该特征项总共有多少个分布值f\_num, 然后一一对应地编号至Num = [0…f\_num]。

## 5问题求解

### 5.1 问题一

#### 5.1.1层次分析法理论介绍

层次分析法AHP (Analytic Hierarchy Process)是美国运筹学家Saaty教授于二十世纪80年代提出的一种实用的多方案或多目标的决策方法。在进行多指标的目标评测或多目标决策时，确定各指标的权数是一个重要而棘手的问题。而权数确定的合理与否，会直接影响到最终的评价、分析结果，层次分析法我我们提供了一套合理确定权数的方案。

层次分析法有以下几个基本步骤：



图1 层次分析法基本步骤

1建立层次模型

在深入分析待研究问题前，将问题中所包含的因素划分为不同层次，将同一层的因素作为比较和评价准则。对层次分析法而言，构建好的结构层次模型对完成好评价任务非常重要。

2构造判断矩阵

层次结构反映了各因素之间的关系，但同一层的因素在目标衡量中所占的比重并不一定相同。在确定某一因素所占的比重时，遇到的主要困难是这些比重通常不易量化。此外，当影响某因素的因子较多时，直接考虑各因子对该因素有多大影响容易导致考虑不周、顾此失彼。为了有效衡量各因子对某一因素的影响程度，Saaty 等人建议可以采取对因子进行两两比较建立成对比较矩阵的办法。

3做一致性检验并计算权重值

设判断矩阵为A，如果A具有完全一致性，则λmax=n，但在实际中构建的判断矩阵很难做到完全一致。一般来说，构造的矩阵只要做到具有相对一致性就能满足需要。计算判断矩阵一致性程度的公式为：

CR=CI/RI

当CR<0.1时，认为判断矩阵A的一致性程度能够被接受。上式中CI=(λmax-n) / (n-1)，RI的取值如表1所示：

表1 RI取值

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| RI | 0.00 | 0.00 | 0.58 | 0.90 | 1.12 | 1.24 | 1.32 | 1.41 | 1.45 |

4根据各模块权重进行综合评分

根据构建的结构模型, 按照上述步骤从上至下对各模块分别进行计算, 得出每一元素对应上一层的权重, 经过逐层计算, 则可计算出最底层的元素对于最上层的权重值。

#### 5.1.2恐怖袭击事件危害性影响因素分析

影响恐怖袭击事件危害性的因素数量繁多，为了从中提取出最重要的一些因素，本文考察了历史上一些著名的恐怖袭击事件[1]，在此基础上提取出了数据库中能较好描述恐怖袭击事件危害性的一些变量：

表2 恐怖袭击事件危害性主要影响因素

|  |  |
| --- | --- |
| **类别** | **子类** |
| **财产损失** | propvalue财产损失价值；propextent财产损害程度 |
| **人员伤亡** | nwound受伤总数；nkill死亡总数 |
| **地域因素** | Region地区；country国家；city城市 |
| **攻击目标** | target具体目标/受害者；target type目标/受害者类型 |

财产损失类：

财产损失价值是一个可以较好反映恐怖袭击事件危害性的因素，比如911事件、马德里爆炸案等恐怖事件都有较大损失金额。

财产损害程度从另一个角度说明了恐怖袭击事件带来的危害。在数据库中，财产损害程度的分级如下表所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 财产损害程度 | 1 | 2 | 3 | 4 |
| 含义 | 灾难性的（可能大于10亿美元） | 重大的（可能大于100万美元，但小于10亿美元） | 较小的（可能小于 100万美元） | 未知 |

人员伤亡类：

受伤总数和死亡总数都是衡量恐怖袭击事件危害性的重要因素，下表给出了一些灾难性恐怖袭击事件的受伤和死亡人数：

|  |  |  |
| --- | --- | --- |
| 事件 | 受伤人数 | 死亡人数 |
| 911事件 | 8191 | 1383 |
| 2014年伊拉克银行袭击案 | - | 1570 |
| 辛贾尔攻势 | - | 953 |
|  |  | 433 |
|  |  |  |

可以看到，这些恐怖袭击案件都有非常高的受伤和死亡人数，且这两个数据并非强正相关关系，因此可以考虑将二者之和一起考虑。

地域类：

关于地域类的特征有非常多，比如地区、国家、所在省份以及城市等。表 列出了地区的表示方法：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 地区编号 | 1 | 2 | 3 | 4 | 5 | 6 |
| 含义 | 北美 | 中美洲和加勒比海地区 | 南美 | 东亚 | 东南亚 | 南亚 |
| 地区编号 | 6 | 8 | 9 | 10 | 11 | 12 |
| 含义 | 中亚 | 西欧 | 东欧 | 中东和北非 | 撒哈拉以南的非洲 | 澳大利亚和大洋洲 |

而省份和城市则以具体的名称和代号来表示。为了更好地将地区信息编码到恐怖袭击事件危害性评估模型中，我们依据国家所在的地区将地区划分成了3个部分，修改后的地区编号如表 所示：

|  |  |  |  |
| --- | --- | --- | --- |
| 修改后的地区编号 | 1 | 2 | 3 |
| 含义 | 其它 | 发展中国家 | 发达国家 |

其中地区编号越大，表明在该地区发生的恐怖袭击事件危害性越大。其中 包含， 包含， 包含， 包含 。

对于城市这一特征，我们将其划分为首都和非首都，本文人为首都地区发生的恐怖袭击案件较非首都地区发生的恐怖袭击案件有更大的危害性。

攻击目标类：

在这一类中，主要考察的特征为target type，原始的target type值如列表 所示：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 原编号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 含义 | 商业 | 政府  [一般意义] | 警察 | 军事 | 流产  有关 | 机场和飞机 | 政府  [外交] | 教育  机构 |
| 原编号 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 含义 | 食物或水供应 | 新闻  记者 | 海事 | 非政府组织 | 其它 | 公民  自身 | 宗教人物/机构 | 电信 |
| 原编号 | 17 | 18 | 19 | 20 | 21 | 22 |  |  |
| 含义 | 恐怖  分子 | 游客 | 运输 | 未知 | 公共  事业 | 暴力  政党 |  |  |

本文认为不同类型目标对象对应的恐怖袭击事件有不一样的危害，因此将上述22类目标对象划分为4类：公共环境类、生活设施类、政府组织类及其它机构类，分别赋值为4,3,2,1。这样target type的新编号和原编号有如下对应关系：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 新编号 | 1 | 2 | 3 | 4 |
| 含义 | 其它机构 | 政府组织 | 生活设施 | 公共环境 |
| 对应的原编号 | 12 13 15 17 20 22 | 2 3 4 7 | 9 11 16 18 19 | 1 5 6 8 10 14 21 |

为了更充分有效地利用这些信息，本文剔除了一些虽然有用但是包含信息量过少的特征，比如具体的财产损失数值。此外，为了减少冗余，本文合并了一些可以合并的数据，比如将受伤总数和死亡总数相加，得到伤亡总数casualty这一特征，对于无法合并的特征，则选取其中最具代表性的特征。以此为原则，本文最终选用的五个特征为：

表3 影响恐怖袭击事件危害性的5个主要特征

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | |
| Propextent | Casualty | Region | Nhostkid | | Target type |
| 财产损害  程度 | 伤亡总数 | 地区 | 人质/绑架  受害者总数 | | 目标/受害者类型 |

对于各个特征进行上述处理后，最后还需要将各个特征进行归一化，这样利用层次分析法求的的权值才有意义。

#### 5.1.3层次分析法求解权值

对问题进行分析讨论后，获得如下的层次结构图：



在确定判断矩阵时，根据Saaty等的建议，用数字1-9及其倒数作为标度，以下为1-9标度的含义:

表4 标度含义

|  |  |
| --- | --- |
| **标度** | **含义** |
| **1** | 表示两个因素相比，具有相同的重要性 |
| **3** | 表示两个因素相比，前者比后者稍重要 |
| **5** | 表示两个因素相比，前者比后者明显重要 |
| **7** | 表示两个因素相比，前者比后者强烈重要 |
| **9** | 表示两个因素相比，前者比后者极端重要 |
| **2,4,6,8** | 表示上述相邻判断的中间值 |

准则层的判断矩阵如下：

表5 准则层判断矩阵

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B1** | **B2** | **B3** | **B4** | **B5** |
| **B1** | 1 | 1/3 | 5 | 2 | 6 |
| **B2** | 3 | 1 | 6 | 3 | 7 |
| **B3** | 1/5 | 1/6 | 1 | 1/3 | 2 |
| **B4** | 1/2 | 1/3 | 3 | 1 | 5 |
| **B5** | 1/6 | 1/7 | 1/2 | 1/5 | 1 |

对上述判断矩阵求得最大特征值及对应的特征向量，进行一致性检验（详见附录程序），得到五个影响因素的层次总排序结果如下：

W=[ 0.2567, 0.4700, 0.0649, 0.1662, 0.0422]

因此可以确定恐怖袭击事件危害性评估模型：

y=XW’

依据上图层次结构图得到的模型可以简洁有效地表示恐怖袭击事件危害性评估模型，但它所考虑的因素毕竟不是那么丰富。为了更充分利用数据库内的信息，减少人为对特征的挑选，本文分析了一个更为复杂的层次结构：



类似地，可以得到准则层的判断矩阵：

表5 准则层判断矩阵

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B1** | **B2** | **B3** | **B4** | **B5** |
| **B1** | 1 | 1/5 | 1/6 | 1/4 | 1/2 |
| **B2** | 5 | 1 | 1/2 | 2 | 2 |
| **B3** | 6 | 2 | 1 | 3 | 4 |
| **B4** | 4 | 1/2 | 1/3 | 1 | 2 |
| **B5** | 2 | 1/2 | 1/4 | 1/2 | 1 |

表5-2-2-3 准则层C判断矩阵2

|  |  |  |
| --- | --- | --- |
| **B1** | **C1** | **C2** |
| **C1** | 1 | 1/3 |
| **C2** | 3 | 1 |

表5-2-2-5 准则层C判断矩阵3

|  |  |  |  |
| --- | --- | --- | --- |
| **B3** | **C7** | **C8** | **C9** |
| **C7** | 1 | 1 | 2 |
| **C8** | 1 | 1 | 2 |
| **C9** | 1/2 | 1/2 | 1 |

表5-2-2-3 准则层C判断矩阵4

|  |  |  |
| --- | --- | --- |
| **B1** | **C1** | **C2** |
| **C1** | 1 | 2 |
| **C2** | 1/2 | 1 |

对上述判断矩阵求得最大特征值及对应的特征向量，进行一致性检验（详见附录程序），得到九个影响因素的层次总排序结果如下：

#### 5.1.4十大恐怖袭击事件预测

利用上节得到的恐怖袭击事件危害性评估模型，可以计算得到近20年来所有有记录的恐怖袭击事件的危害性，其中危害性最高的10件事件及其对应的y值如表所示：

表6 近20年危害性最高的十大恐怖袭击事件

|  |  |  |
| --- | --- | --- |
| **序号** | **事件编号** | **Y值** |
| 1 | 200109110005 | 98.0268 |
| 2 | 199808070002 | 43.3371 |
| 3 | 201603080001 | 15.4353 |
| 4 | 200802010006 | 11.7738 |
| 5 | 200409010002 | 11.0335 |
| 6 | 200607120001 | 10.5376 |
| 7 | 200708150005 | 10.4867 |
| 8 | 200708160008 | 10.4867 |
| 9 | 201710010018 | 9.2163 |
| 10 | 201710140002 | 9.1552 |

为了分析预测结果是否合理，本文根据年、月、日和“恐怖袭击”四个关键词在搜索引擎搜索相关信息，均能找到对应的事件。此外，本文预测的十大恐怖袭击事件和某网站给出的十大知名恐怖袭击事件存在一定交集。上述两点可以说明本文的预测结果较为合理。

表7 近20年危害性最高的十大恐怖袭击事件的内容

|  |  |
| --- | --- |
| 事件编号 | 事件描述 |
| 200109110005 | **美国911事件** |
| 199808070002 | **美国驻肯尼亚和坦桑尼亚的大使馆遭到汽车炸弹的袭击** |
| 201603080001 |  |
| 200802010006 |  |
| 200409010002 | **俄罗斯 别斯兰人质事件** |
| 200607120001 | 印度孟买恐怖连环袭击 |
| 200708150005 | 伊拉克发生严重的爆炸袭击事件 |
| 200708160008 | 伊拉克北部尼尼微省发生4起自杀式汽车炸弹袭击事件，这是自伊拉克战争爆发以来造成人员伤亡最为惨重的袭击事件 |
| 201710010018 | 拉斯维加斯发生枪击事件，造成至少59人丧生、527人受伤，刷新美国现代史上“最严重枪击案”纪录 |
| 201710140002 | 索马里首都摩加迪沙市区一家酒店附近发生爆炸袭击事件。汽车炸弹袭击已造成至少358人死亡、228人受伤、56人失踪 |

表8 某网站给出的十大知名恐怖事件

|  |  |
| --- | --- |
| **序号** | **事件描述** |
| 1 | **2001年美国911事件** |
| 2 | **2004年俄罗斯别斯兰人质事件** |
| 3 | 1983年贝鲁斯特事件 |
| 4 | 2004年伊拉克清真寺连环爆炸案 |
| 5 | 1993年孟买爆炸案 |
| 6 | 1988年洛克比空难 |
| 7 | **1998达累斯萨拉姆的大使馆爆炸案** |
| 8 | 2002年印度尼西亚巴厘岛爆炸案 |
| 9 | 2004年马德里爆炸案 |
| 10 | 1995年俄克拉荷马城爆炸案 |

从判断矩阵（表）及求的的权值（公式）不难发现，上述模型对财产损失和人员伤亡两个指标更为重视；如果对地区这一指标更加重视，即赋予地区之一特征更大的权值，本文得到了以下的十大恐怖袭击事件预测结果：

|  |  |
| --- | --- |
| 事件编号 | 事件描述 |
| 200109110005 | **美国911事件** |
| 199808070002 | **美国驻肯尼亚和坦桑尼亚的大使馆遭到汽车炸弹的袭击** |
| 201603080001 |  |
| 200607120001 | 印度孟买恐怖连环袭击 |
| 200708150005 | 伊拉克发生严重的爆炸袭击事件 |
| 200708160008 | 伊拉克北部尼尼微省发生4起自杀式汽车炸弹袭击事件，这是自伊拉克战争爆发以来造成人员伤亡最为惨重的袭击事件 |
| 200802010006 |  |
| 200409010002 | **俄罗斯 别斯兰人质事件** |
| 201710010018 | 拉斯维加斯发生枪击事件，造成至少59人丧生、527人受伤，刷新美国现代史上“最严重枪击案”纪录 |
| 201710140002 | 索马里首都摩加迪沙市区一家酒店附近发生爆炸袭击事件。汽车炸弹袭击已造成至少358人死亡、228人受伤、56人失踪 |

类似地，如果更加关注受害者类型，得到的十大恐怖袭击事件预测结果则为：

|  |  |
| --- | --- |
| 事件编号 | 事件描述 |
| 200109110005 | **美国911事件** |
| 199808070002 | **美国驻肯尼亚和坦桑尼亚的大使馆遭到汽车炸弹的袭击** |
| 201603080001 |  |
| 200607120001 | 印度孟买恐怖连环袭击 |
| 200708150005 | 伊拉克发生严重的爆炸袭击事件 |
| 200708160008 | 伊拉克北部尼尼微省发生4起自杀式汽车炸弹袭击事件，这是自伊拉克战争爆发以来造成人员伤亡最为惨重的袭击事件 |
| 200409010002 | **俄罗斯 别斯兰人质事件** |
| 200802010006 |  |
| 201710010018 | 拉斯维加斯发生枪击事件，造成至少59人丧生、527人受伤，刷新美国现代史上“最严重枪击案”纪录 |
| 201710140002 | 索马里首都摩加迪沙市区一家酒店附近发生爆炸袭击事件。汽车炸弹袭击已造成至少358人死亡、228人受伤、56人失踪 |

不难发现，表 至表 的内容都是一样的，只不过顺序稍有改变。这表明本文选取的这些特征均能充分体现恐怖袭击案件的危害性，同时，这也揭示了这样一个规律——危害性较高的恐怖袭击事件几乎同时具备以下特点：

a具有较高的伤亡人数；

b发生在一些相对较重要的地理位置，比如一个国家的政治、经济中心等；

c被攻击对象多处于人流密集的公共环境中。

#### 5.1.5部分恐怖袭击事件分级结果

根据第3节得到的恐怖袭击事件危害性评估模型，对表9中十大典型事件的分级结果如下：

表9 十大典型时间的分级结果

|  |  |  |
| --- | --- | --- |
| **事件编号** | **Y值** | **危害级别** |
| 200108110012 | 4.5667 | 1 |
| 200511180002 | 1.9990 | 1 |
| 200901170021 | 0.1207 | 2 |
| 201402110015 | 0.0968 | 3 |
| 201405010071 | -0.0560 | 5 |
| 201411070002 | 0.1902 | 2 |
| 201412160041 | 1.4724 | 1 |
| 201508010015 | -0.0356 | 4 |
| 201705080012 | 0.6573 | 2 |

考虑到带来重大危害的恐怖袭击案件在所有恐怖袭击案件中所占的比例还是较低，因此本文假设恐怖袭击案件的等级划分应服从下图所示的分布规律：



结合所有恐怖袭击事件的y值，得到如下量化分级模型：

|  |  |  |  |
| --- | --- | --- | --- |
| y值 | y≥1.0343 | 1.0343＞Y≥0.1207 | 0.1207＞Y≥-0.0152 |
| 等级 | 1 | 2 | 3 |
| y值 | -0.0152＞y≥-0.00458 | -0.0458＞y |  |
| 等级 | 4 | 5 |  |

### 5.2 问题二

#### 5.2.1 问题二数据预处理

数据预处理包含数据清洗和特征处理两个部分。

1.数据清洗。数据清洗旨在将原始数据进行全局清洗。其主要目的是消除数据中的错误、冗余和噪声，确保数据的完整性和标准数值化，从而有利于下一步的数据分析。数据清洗的步骤包括数据剔除，数据完整化，数据数值化和数据标准化。这些步骤的详细意义和操作如下文所述。

**数据剔除：**数据剔除的目的是将原始数据中与所有任务都不相关的错误与冗余数据全部去除。分为两个方向：

（一）纵向特征剔除。剔除与所有任务都不相关的纵向特征列。具体包含了表示数据来源的特征项（scite1, scite2,scite3,dbsource），与数值编码重复了的描述性语句的特征项（如targtype1\_txt 与targtype1意义重复了，且targtype1已经数值化了，所以targtype1\_txt可以删除），其他的一些不方便使用的描述性语句（如location, summary等）。具体见表5-2-1-1

表5-2-1-1

|  |  |
| --- | --- |
| 处理方法 | 对应属性列表 |
| 直接删除，  完全不相关属性 | approxdate,provstate,gsubname,gname2,gsubname2,gname3,gsubname3,divert,kidhijcountry,corp3,target3,resolution,scite1,scite2,scite3,dbsource' |
| 直接删除，  数据过于稀疏 | crit1,crit2,crit3,corp1,target1,corp2,target2 |
| 可删除，  描述性语句 | addnotes,ransomnote,propcomment,weaptype1, weaptype1\_txt, weapsubtype1\_txt,weaptype2\_txt, weapsubtype2\_txt, weaptype3\_txt, weapsubtype3\_txt, weaptype4\_txt, weapsubtype4\_txt,propextent\_txt,hostkidoutcome\_txt,country\_txt,alternative\_txt,attacktype1\_txt, attacktype2\_txt, attacktype3\_txt, targtype1\_txt,targsubtype1\_txt,natlty1\_txt, targtype2\_txt, targsubtype2\_txt, natlty2\_txt, targtype3\_txt, targsubtype3\_txt,natlty3\_txt,claimmode\_txt,claimmode2\_txt,claimmode3\_txt,region\_txt,location, summary, weapdetail |

(二) 横向样本剔除。将基本上认为该事件不是恐怖主义行为的样本删除。即如果某事件的疑似恐怖主义（doubtterr）标记为‘1’，则表示基本上不怀疑该事件是恐怖主义行为，所以该样本属于错误样本，可以直接删除，这样更加有利于下一步的数据分析。

**数据完整化：**所有特征项中存在数据缺失的，若该列数据中有‘Unknown’项则空缺项标记为‘unknown’，若该列数据中没有有‘Unknown’项则空缺项标记为‘NaN’。

**数据数值化：**由于在后续数据分析中主要采用的是xgboost算法，而且本案例中的非数值数据是离散的，没有数值大小意义的，所以这里的数值化编码可以采用二进制编码格式。具体做法为：

对于非数值数据，如一个单词或一段描述性语句的字符串，先统计该特征项总共有多少个分布值f\_num, 然后一一对应地编号至Num = [0…f\_num], 再按照二进制编码格式将Num编码转化为Binary\_Num。

**数据标准化**：对于‘unknown’项，统一转化为数值‘-99’。

2. 特征处理

特征处理包含数值处理、one-hot二进制编码、时间处理和文本处理。

**数值处理：**

(一) 幅度调整/归一化：

使用两种方法将数据进行归一化：

**Min-max 归一化：**



min-max标准化方法是对原始数据进行线性变换。设minA和maxA分别为属性A的最小值和最大值，将A的一个原始值x通过min-max标准化映射成在区间[0,1]中的值x’，其公式为:

新数据=(原数据-最小值)/(最大值-最小值)

**z-score 标准化：**

这种方法基于原始数据的均值(mean)和标准差(standard deviation)进行数据的标准化。将A的原始值x使用z-score标准化到x’。z-score标准化方法适用于属性A的最大值和最小值未知的情况，或有超出取值范围的离群数据的情况。将数据按其属性(按列进行)减去其均值，然后除以其方差。最后得到的结果是，对每个属性/每列来说所有数据都聚集在0附近，方差值为1。



其中μ为数据的均值，δ为方差。

（二）统计值：包括max, min, mean, std等。使用pandas库序列化数据后，可以得到数据的统计值。

（三）离散化：把连续值转成非线性数据。例如从0.03到100的连续值，用99维的向量代表每一个维度所处的区间，即1.2元和1.6的向量都是 [0,1,0,…,0]。

**One-hot二进制编码：**

由于在后续数据分析中主要采用的是xgboost算法，而且本案例中的非数值数据是离散的，没有数值大小意义的，所以这里的数值化编码可以采用二进制编码格式。具体做法为：

对于非数值数据，如一个单词或一段描述性语句的字符串，先统计该特征项总共有多少个分布值f\_num, 然后一一对应地编号至Num = [0…f\_num], 再按照二进制编码格式将Num编码转化为Binary\_Num。

**时间处理：**

使用calendar的weekday方法将iyear, imonth, iday三列数据转换为星期数据并构造’xingqi’列，将日期转换为星期。例如iyear=2015,imonth=1,iday=1构造xingqi=5。

**文本处理：**

（一）词袋：文本数据预处理后，去掉停用词，剩下的词组成的list，在词库中的映射稀疏向量。

（二）把词袋中的词扩充到n-gram：n-gram代表n个词的组合。

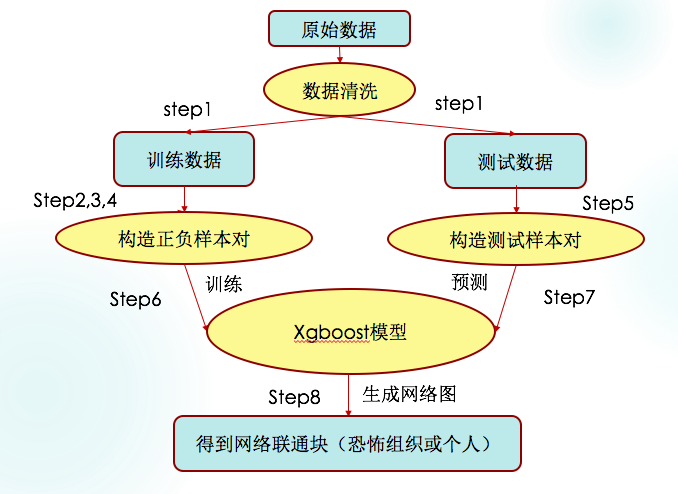
（三）使用TF-IDF特征：TF-IDF是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。TF(t) = (词t在当前文中出现次数) / (t在全部文档中出现次数)，IDF(t) = ln(总文档数/ 含t的文档数)，TF-IDF权重 = TF(t) \* IDF(t)。

#### 5.2.2 问题二模型建立

本文受启发于人脸识别领域中的构造正负样本进行分类学习的方法，巧妙地将问题二转化处理为二分类结合连同块的问题，并结合python工具包xgboost进行训练与预测。总体算法框架及其具体详细步骤见下文描述。

##### 5.2.2.1总体算法框架

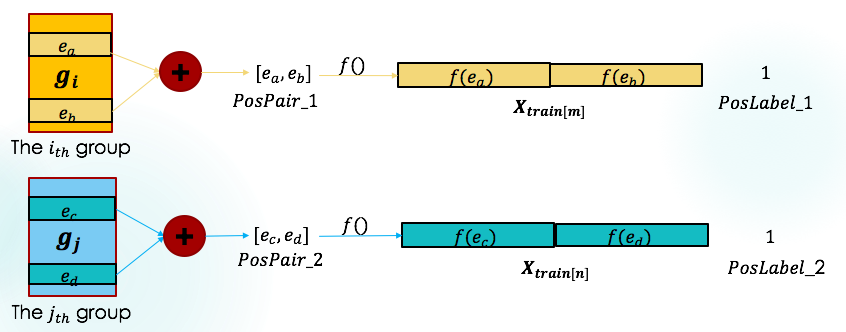
任务二的总体算法结构图如图\*\*所示，其中蓝色方框代表输入输出和中间的数据结果，黄色椭圆代表算法操作。每一个算法操作的详细步骤可参考下一小节对step1-step9的详细介绍，也可结合附录中的任务二代码来深入理解。



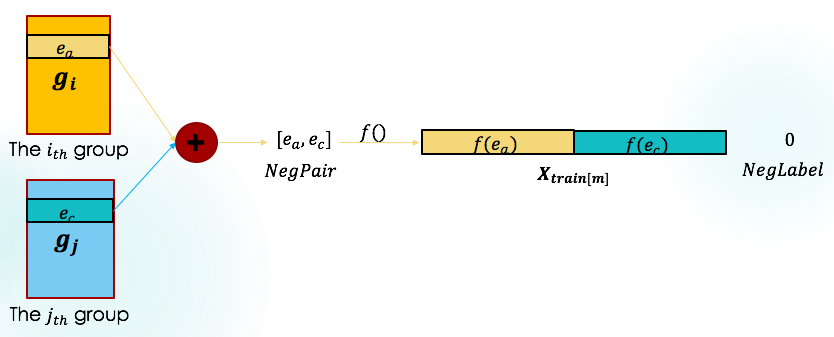
图\*\*：

##### 5.2.2.2详细步骤

任务二的详细求解步骤step1-step9详见下文所述，也可与附录中的任务二代码相结合理解。



图\*： 如何构造正样本对，类别标签。其中表示第个恐怖组织，表示第个恐怖组织; 和表示恐怖组织做过的某两件恐怖事件，和表示恐怖组织做过的某两件恐怖事件；表示特征索引，即表示恐怖事件所对应的特征向量。

**

图\*： 如何构造负样本对，类别标签。其中表示第个恐怖组织，表示第个恐怖组织; 表示恐怖组织做过的某件恐怖事件，表示恐怖组织做过的某件恐怖事件；表示特征索引，即表示恐怖事件所对应的特征向量。

step1: 对清洗后的数据拆分为训练集train和测试集test。其中训练集train为所 有年份的有明确恐怖组织归属（即gname不为‘unknown’）的所有恐怖事件的特征矩阵；测试集test为2015-2016年间的无恐怖组织归属（即gname为 ‘unknown’）的所有恐怖事件的特征矩阵。

step2: 在训练集上，统计各个恐怖组织所对应的恐怖事件. 获得相应的geList, 其第i行为：

其中，代表的是group\_id 为 i 的恐怖组织所做过的所有恐怖事件的集合。

step3: 在训练集上构造正样本对及其对应标, 构造负样本对及其对应标. 具体构造方式如图\*\*所示。

Step4:将所有训练样本对（）, 通过 索引得到其特征向量，并按照图\*\*所示两两连接起来，组成最终的训练数据, 其中表示最终输入Xgboost 的训练集特征矩阵， 表示其对应的类别标签。

Step5:在测试集上构造测试样本对及其对应标并通过 索引得到其特征向量，并按照图\*\*所示两两连接起来，组成最终的测试数据. 表示最终输入Xgboost 的测试集特征矩阵。

Step6:将step4所获的的训练数据输入Xgboost训练器进行二分类训练，得到模型。

Step7:将step5所获的的测试数据输入step6 训练所得的，得到预测的标签。

Step8:将作为网络的节点对，作为连接边（1代表两节点之间有一条边连接，0代表没有边连接），应用python包networkx绘制出网络，并得到多个联通块:

每一个联通块表示一个可能的恐怖组织或个人。

Step9:对每个，结合任务一的恐怖事件危害性评价模型，计算恐怖组织的危害性：

其中表示属于恐怖组织的恐怖事件。

#### 5.2.3 问题二模型求解

##### 5.2.3.1xgboost原理

1. Tree ensemble

Boosting Tree 最基本的部分是回归树（CART）， CART会把输入根据属性分配到各个也子节点上，而每个叶子节点上面会对应一个分数值。回归树可以做很多事情，比如概率预测，排序等等。

假设有两颗回归树，则两棵树融合后的预测结果用公式表示为：

其中,K 是树的个数, 是第k棵树对于输入输出的得分，是相应函数，则目标函数为：

函数描述之间的距离。

1. XGBoost

更加一般地,对于一般损失函数，XGBoost会使用泰勒展开的形式进而用到二阶导数。

目标函数：

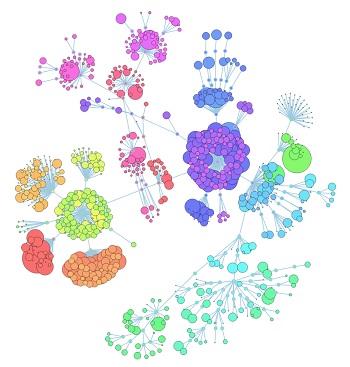
用泰勒展开来近似目标函数：

有了这个更新后的目标函数后，可以看出这个目标函数仅仅依赖一阶似然的一阶和二阶导数。对于Adaboost，通常只能用用平方损失函数和通过拟合残差的方式来求解。GBDT在Adboost基础上做了一部优化，用释然函数在每一个树处的梯度来近似残差。有了上述通用的形式后，就可以使用任意形式的损失函数。

##### 5.2.3.2实验结果

1. 各组织聚类结果

对所有测试样本构成的网络图可视化得图\*\*。



图\*\*

结果分析：

1. 由图\*\*可以看出，的确有一些大小组织的迹象，说明本文算法所训练的分类起到了一定的效果，能有效地判别出对划分恐怖组织有效的特征，并主要根据这些特征在‘unknown’类型上做出分类的预测；
2. 但可能依然存在一些噪声点或者是没有很有效预测的点，使得各组织之间存在联通性。这需要进一步的对模型挑参，清洗整理数据，增加训练次数，从而增加模性的准确性。
3. 危害性最高的5类

结合任务一得到的恐怖事件危害性评价模型，对每个恐怖组织，统计其涉及的恐怖事件的危害性之和，最终计算恐怖组织的危害性：

1. 其中表示属于恐怖组织的恐怖事件。
2. 计算得前五名，编码得到结果如表\*\* 所示：

**表2 前六名恐怖组织**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 恐怖组织 | 1号 | 2号 | 3号 | 4号 | 5号 | 6号 |
| 组织危害性 | 566.48 | 322.65 | 199.90 | 124.01 | 65.78 | 21.70 |

三、 嫌疑排序

**表2 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 |
| 201701090031 | 1 | 3 | 4 | 5 | 2 |
| 201702210037 | 3 | 2 | 1 | 5 | 4 |
| 201703120023 | 2 | 1 | 3 | 4 | 5 |
| 201705050009 | 1 | 3 | 4 | 2 | 5 |
| 201705050010 | 4 | 2 | 1 | 5 | 3 |
| 201707010028 | 2 | 1 | 3 | 5 | 4 |
| 201707020006 | 1 | 2 | 5 | 3 | 4 |
| 201708110018 | 3 | 2 | 4 | 1 | 5 |
| 201711010006 | 1 | 2 | 4 | 3 | 5 |
| 201712010003 | 2 | 3 | 1 | 4 | 5 |

#### 5.2.4 问题二深入分析

1. 数据准备

由于数据集中的时间截至到2017年7月，故近三年的恐怖袭击事件的时间取自2014到2016年期间。

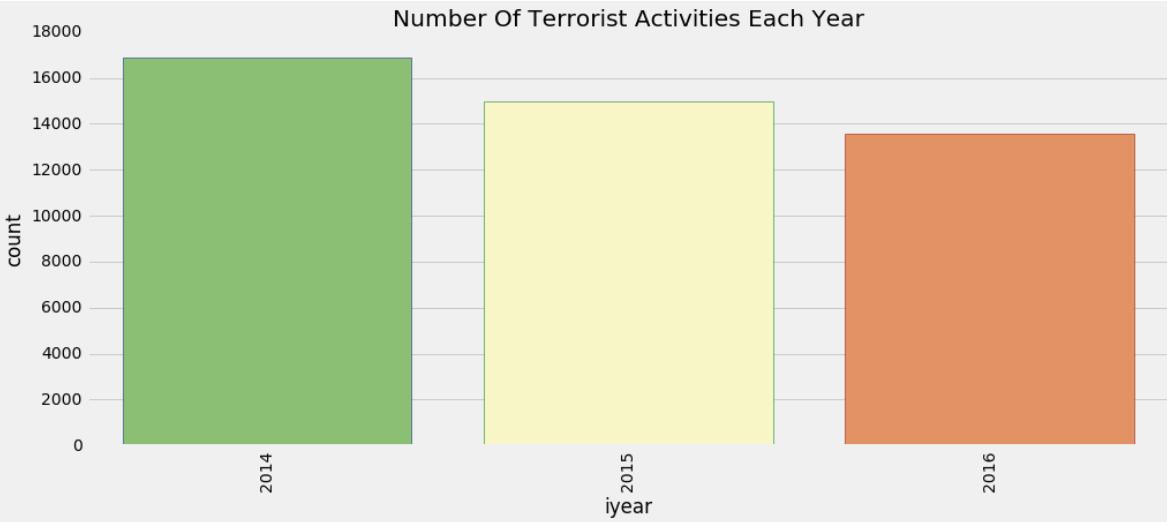
1. 基础分析

根绝2014年到2016年的恐怖袭击案件统计得到：

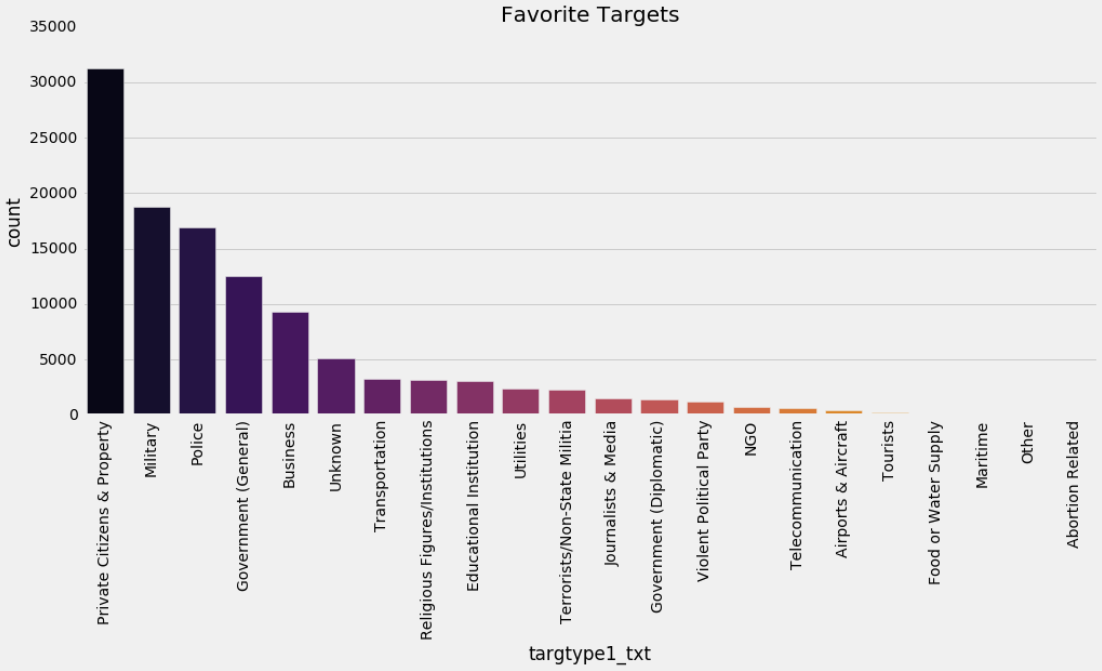
受到恐怖袭击次数最多的国家: Iraq

受到恐怖袭击次数最多的地区: Middle East & North Africa

在一次恐怖袭击中死亡人数最多的事件: 1570.0 that took place in Iraq

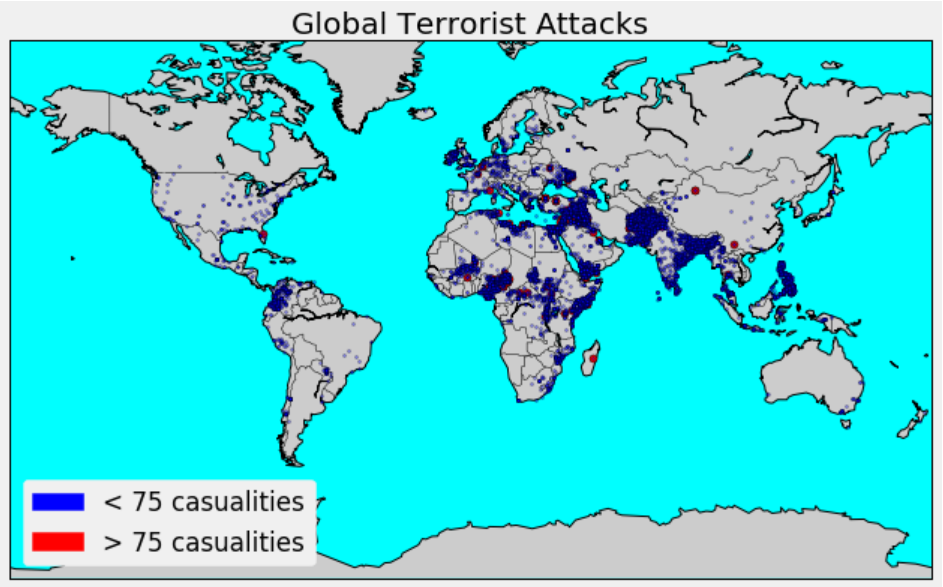


虽然自二十一世纪初的恐怖袭击态势呈现直线上升事态，但就2014至2016三年的数据分析可以看出近三年的恐怖案件数量是稳中有降的。



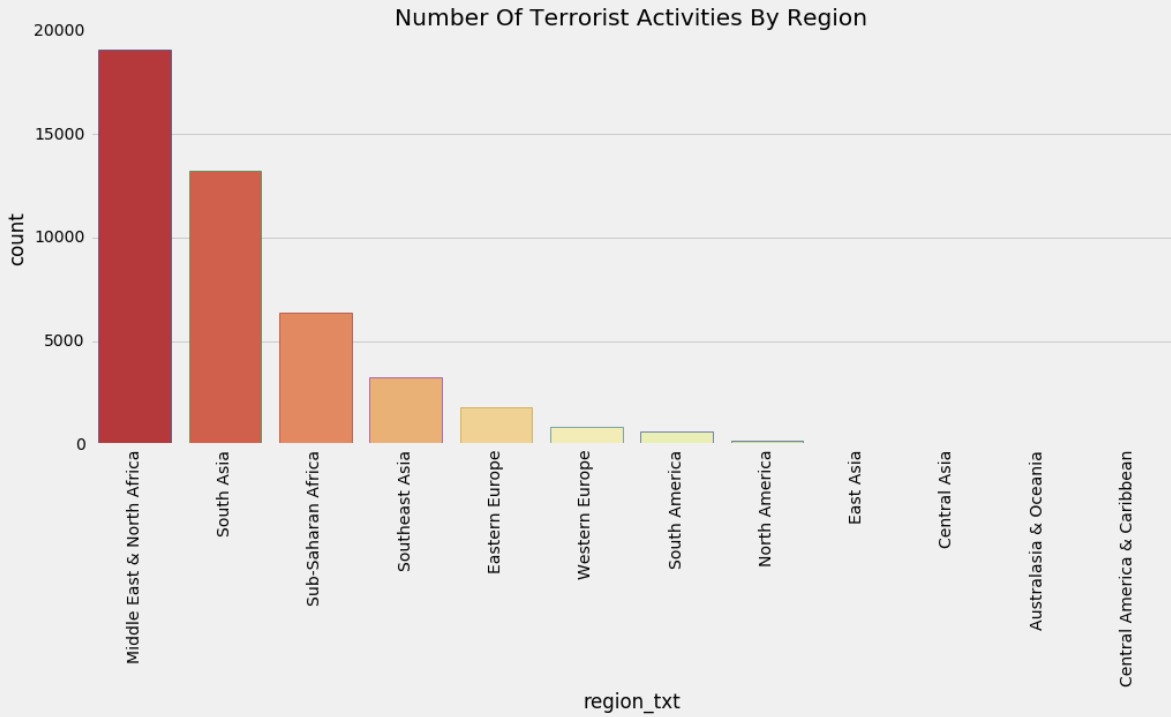
近三年恐怖袭击案的目标类型。

1. 全球性恐怖袭击

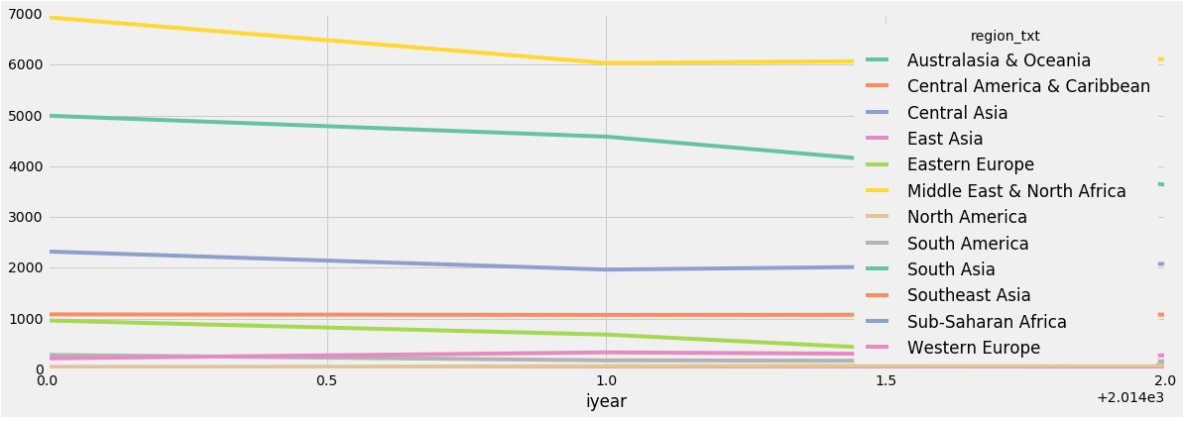


从全球的角度来看，近三年来小于75人伤亡的案件主要发生在，超过75人的案件发生在。

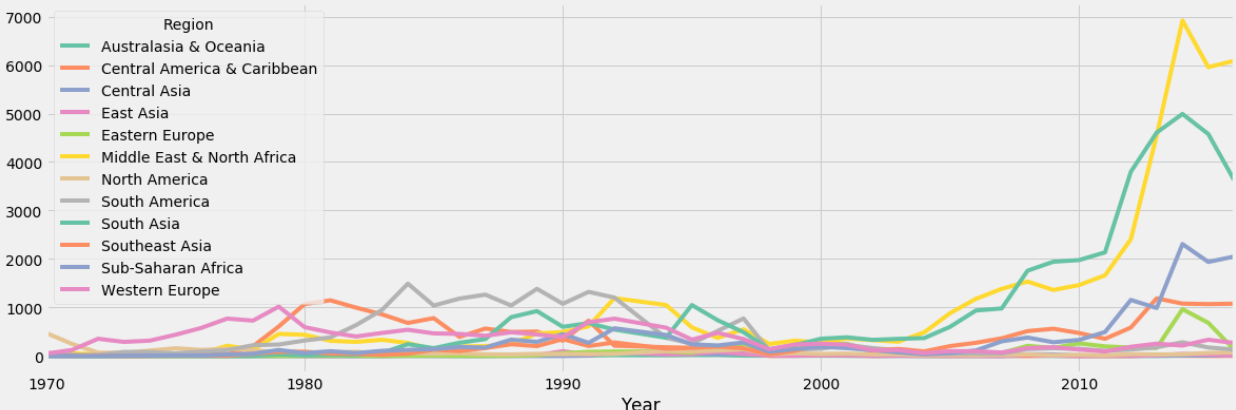
1. 地域恐怖案件



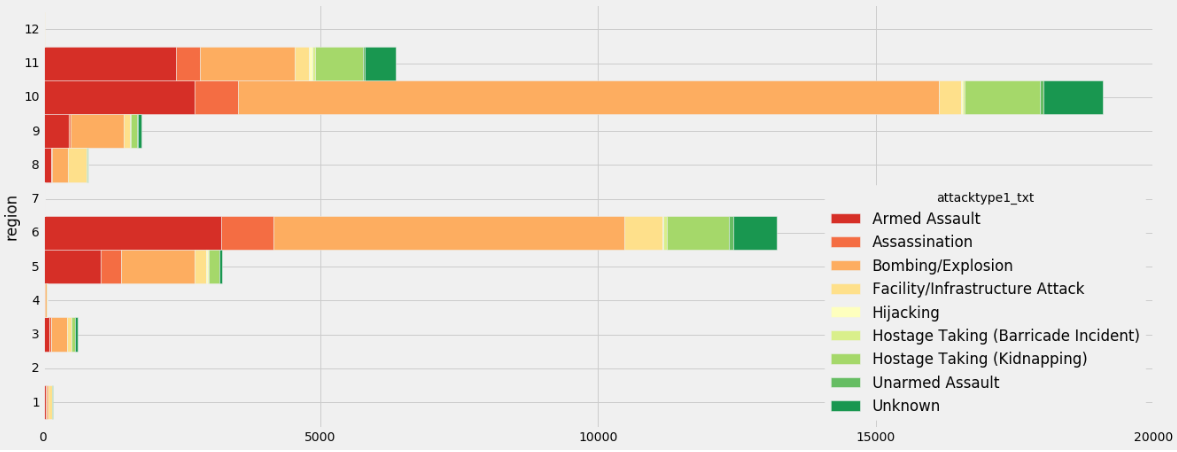
近三年恐怖袭击案的主要地区。



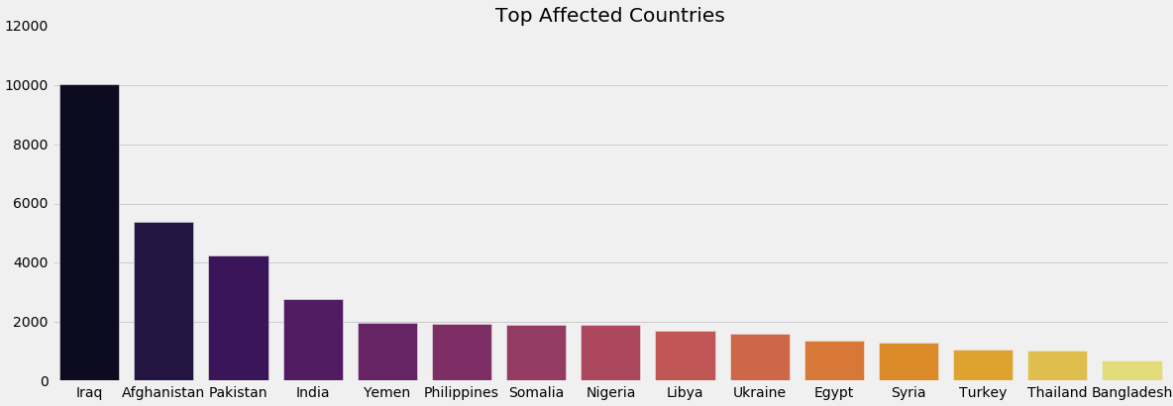
近三年各个地区的恐怖袭击案的数量变化。



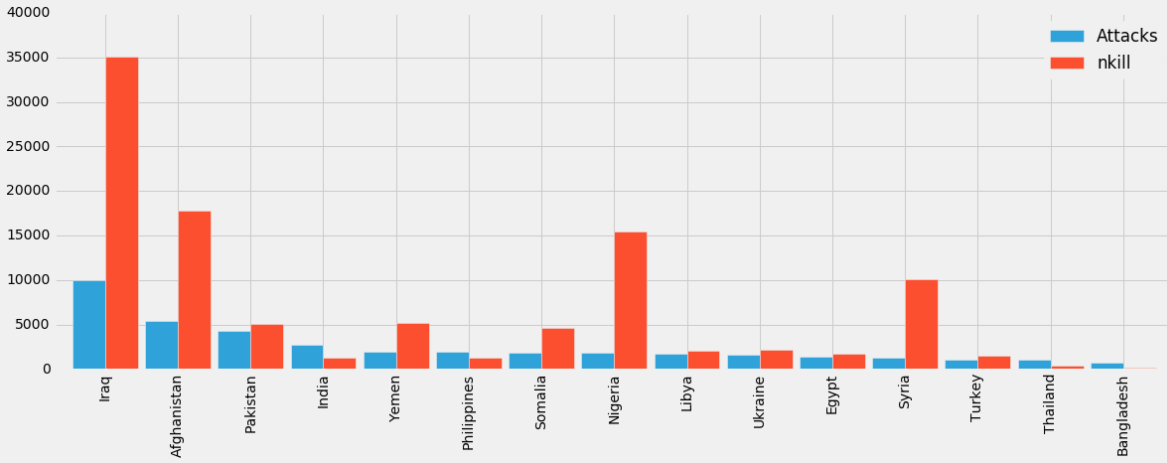
1970年以来各个地区的恐怖袭击案的数量变化。



近三年来恐怖袭击案的攻击对象。

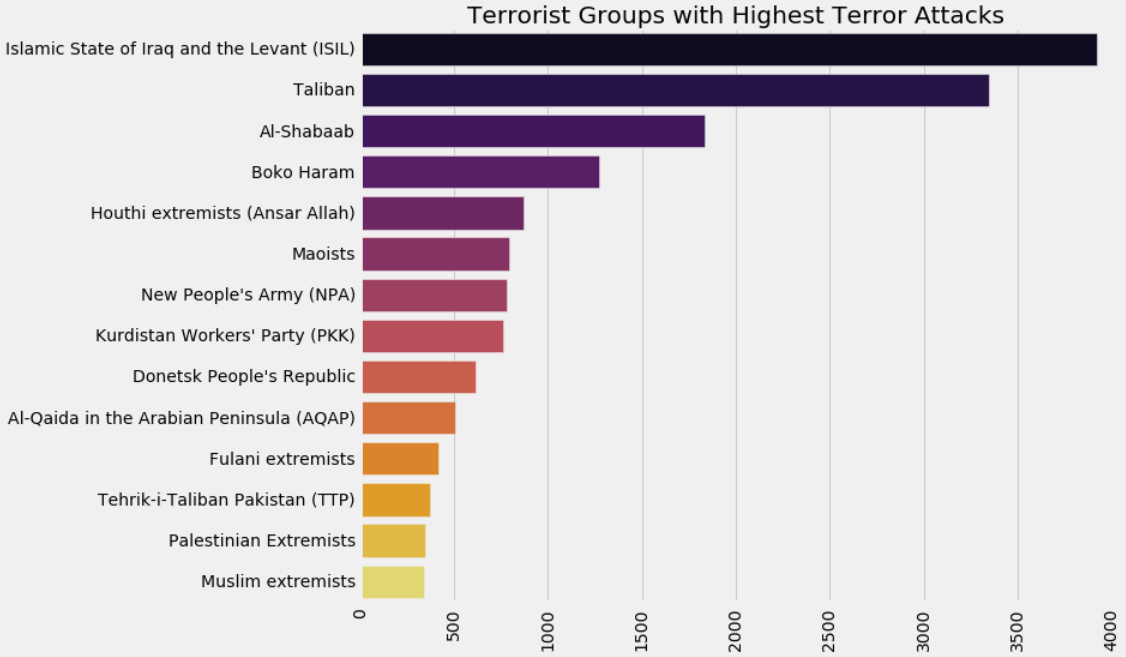


近三年来发生案件次数最多的国家（受恐怖影响最大的国家）

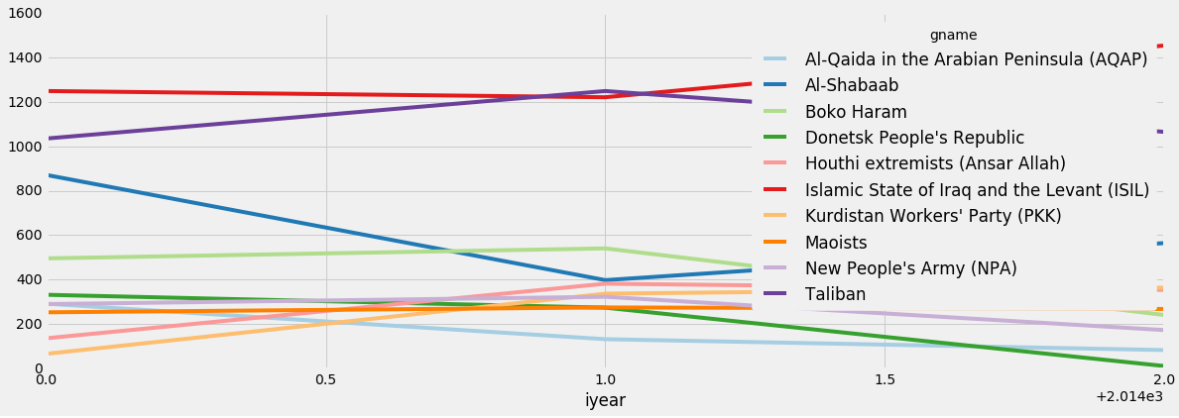


近三年来在主要国家发生案件次数和死亡人数。

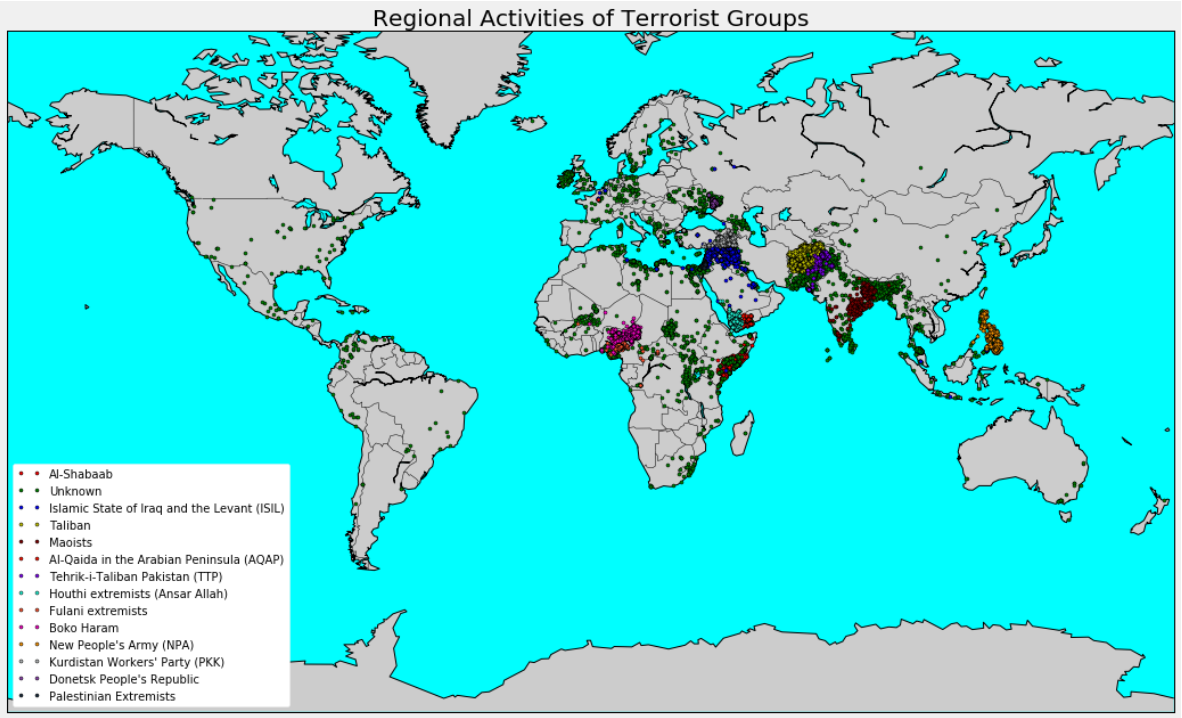
1. 恐怖组织



近三年来实施策划或宣称恐怖袭击案件负责的恐怖组织。

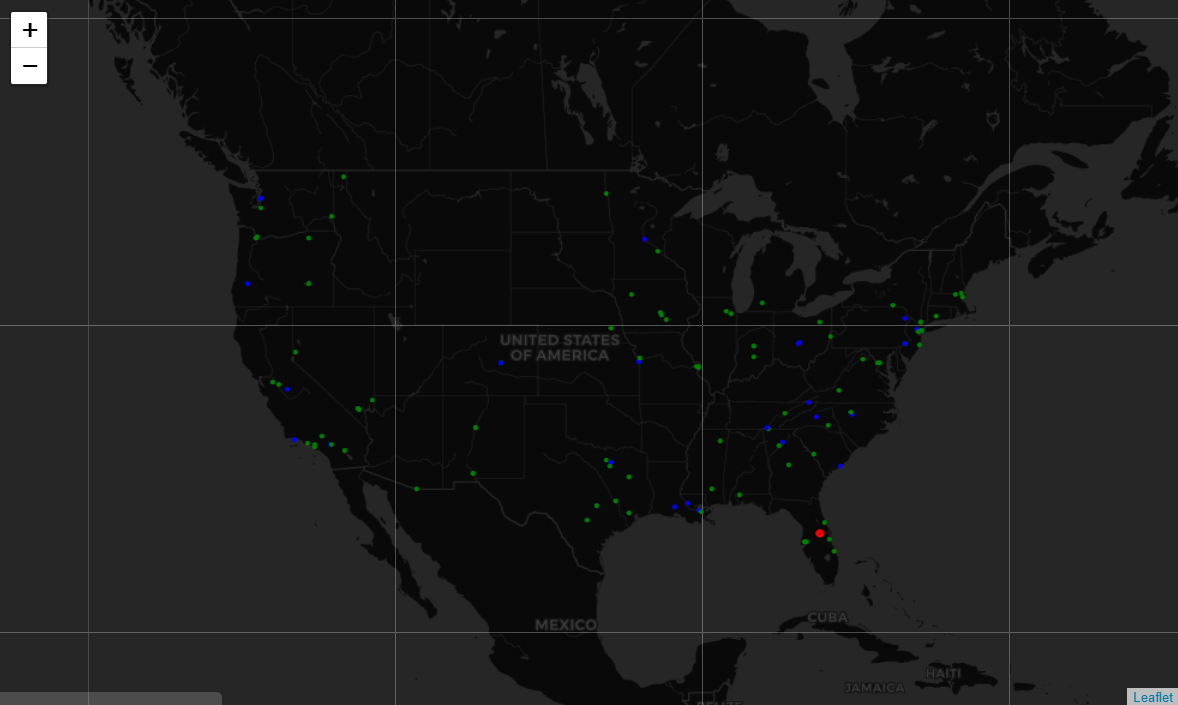


近三年来实施策划或宣称恐怖袭击案件负责的恐怖组织和时间关系。

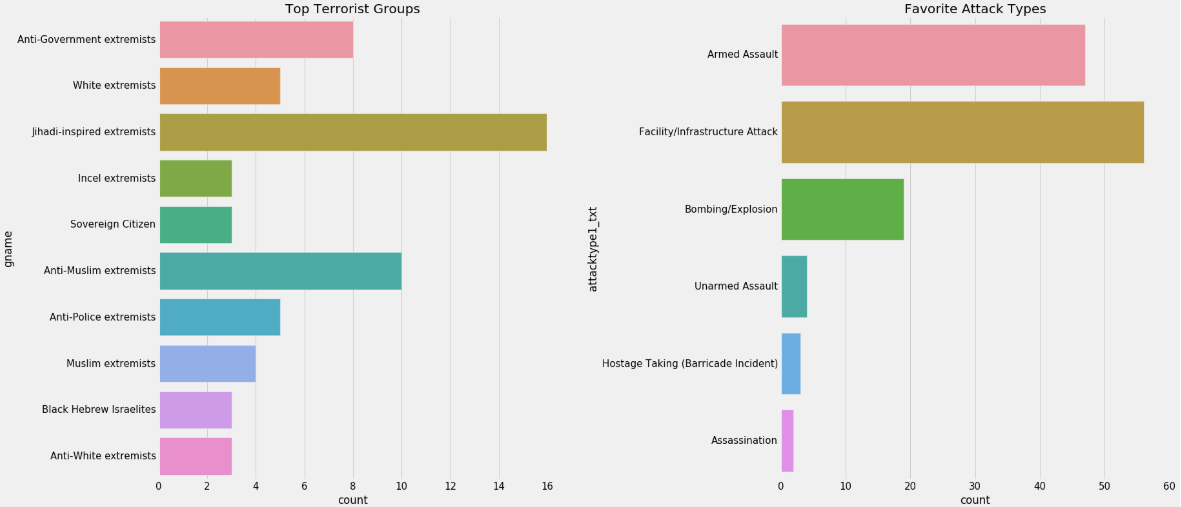


近三年来实施策划或宣称恐怖袭击案件负责的恐怖组织和地域关系。

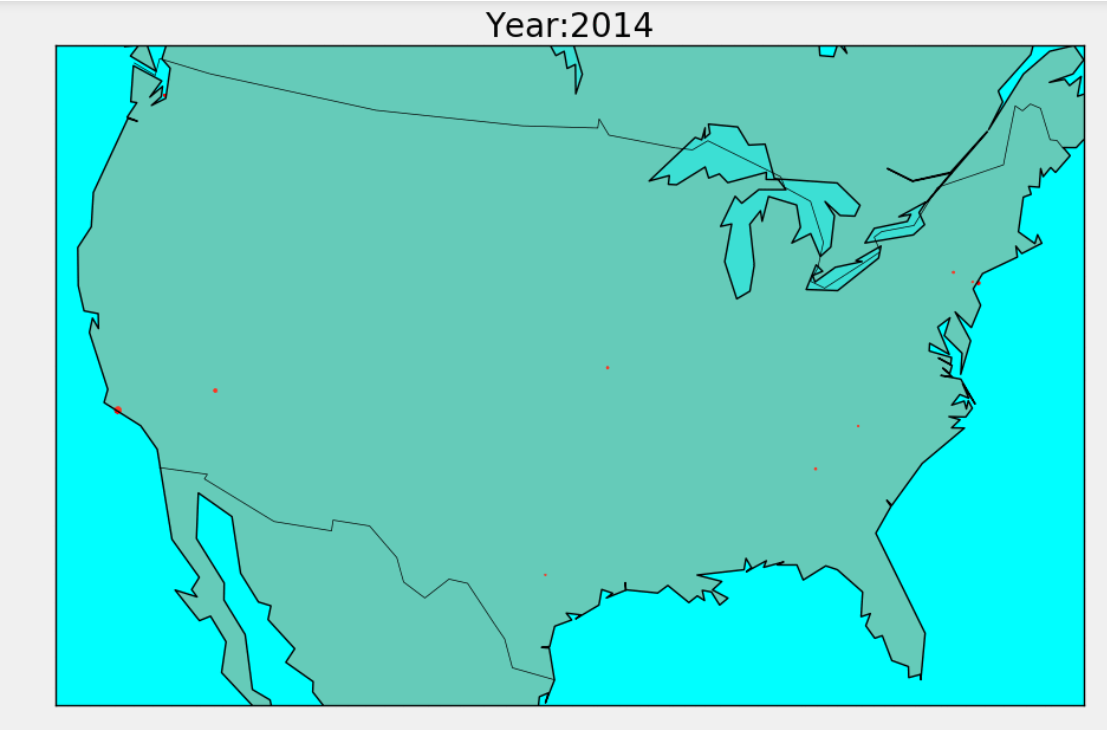
1. 美国恐怖案件

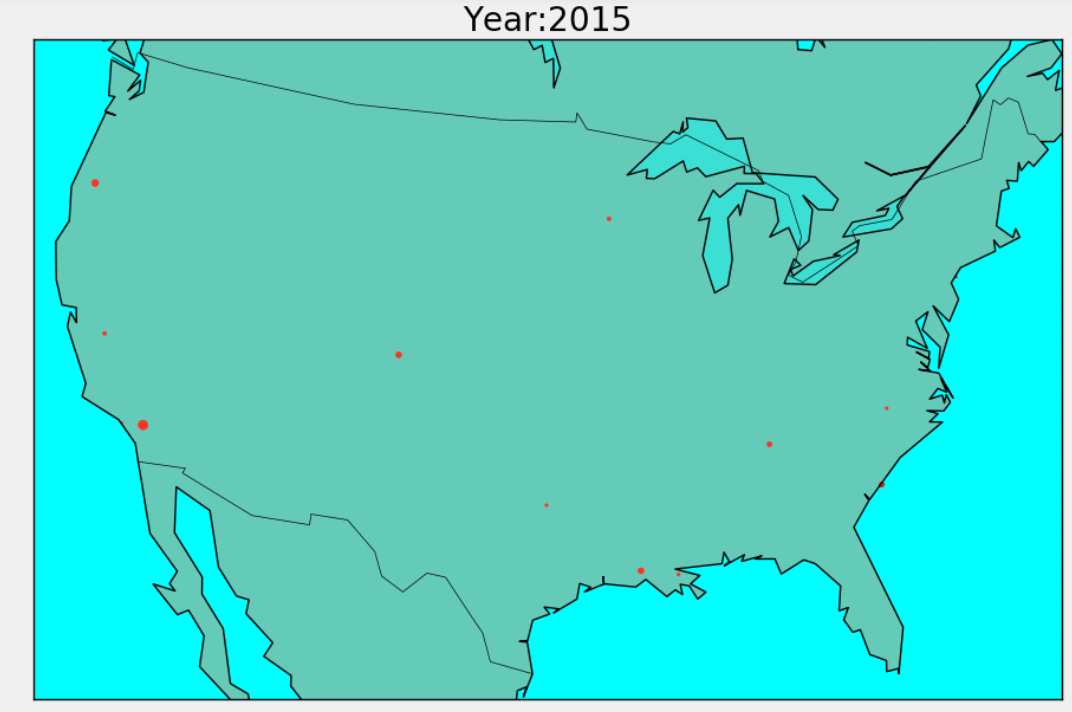


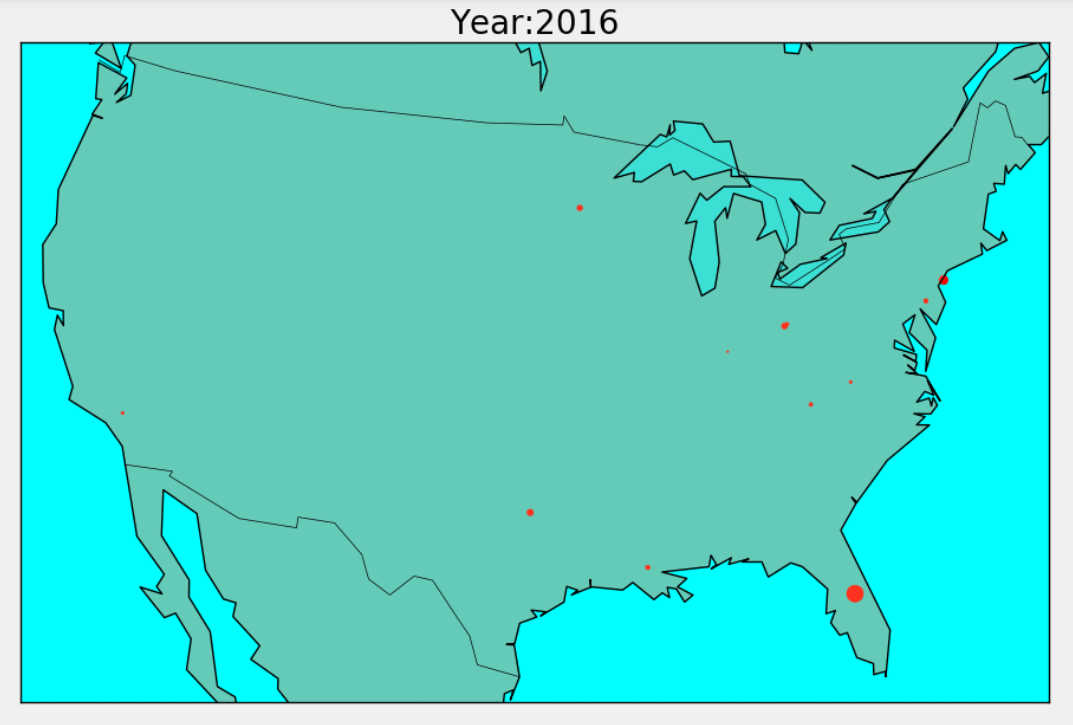
近三年来在美国发生的恐怖袭击案件（按伤亡程度划分）



近三年来美国恐怖袭击案件的组织和攻击目标。

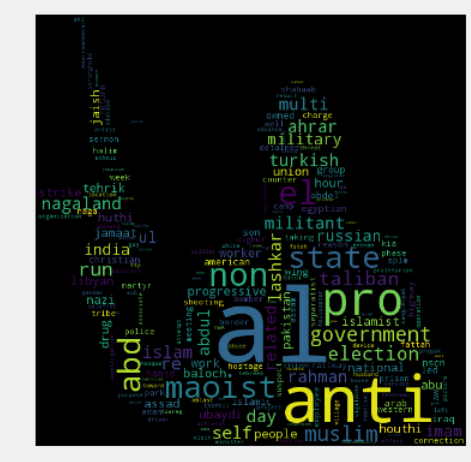






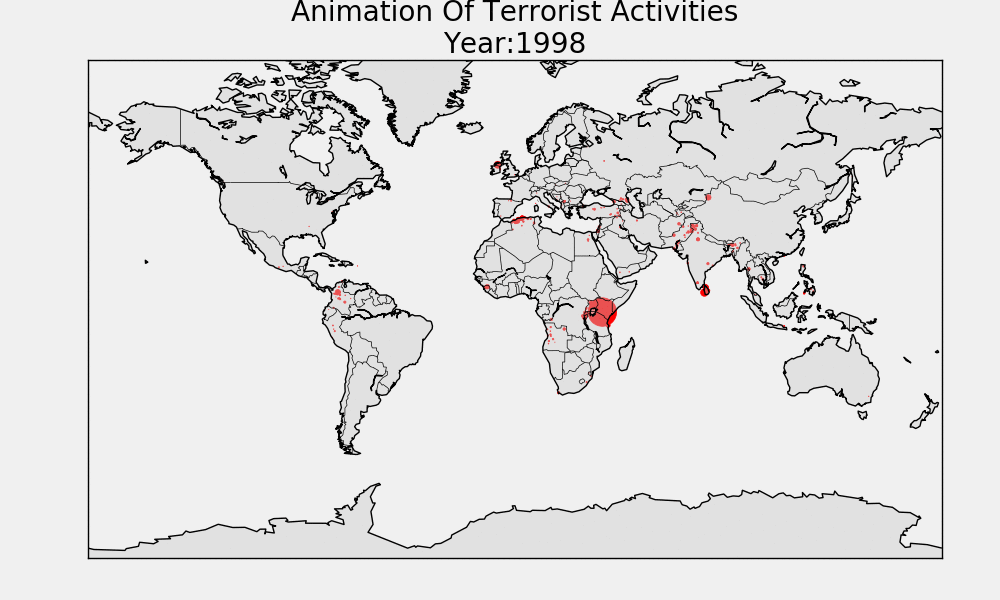
近三年美国每年发生的恐怖袭击案件变化图。

1. 恐怖袭击背后的动机



根据近三年的美国恐怖袭击案的相关关键词的词云分析作案动机。

1. 恐怖活动全球性蔓延



动图。从1998年到2017年的恐怖袭击案件。

最后：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。

### 问题三

**六、模型优缺点及推广**

**七、参考文献**

[1] 姜启源、谢金星，《数学模型》，北京：高等教育出版社，2007

[2] Friedman, Jerome H., and Jacqueline J. Meulman. "Multiple additive regression trees with application in epidemiology." *Statistics in medicine* 22.9 (2003): 1365-1381.

[3]

[4]

[5]

# 八、代码附录

## 任务二

### 任务二数据清洗

## 任务二解决算法

from numpy import loadtxt

from xgboost import XGBClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

import pandas as pd

import numpy as np

import itertools

import networkx as nx

# load data

terror=pd.read\_csv('/Users/alibaba/gitlab/2018-mathmodels/q2/Q2\_1.csv',encoding='ISO-8859-1')

print(terror[:4])

# 提取第二问的所有特征列，并转换成 list

gname = terror['gname\_10encode'].as\_matrix()

print(gname[:10])

# seperate train and test

# test: 2015-2016, gname = 'unknown'

# train: 除了test 集的都是。考虑是否拿一点出来做validation。

# cleaned\_dataset 第一列是event\_id, 倒数第一列是goup\_id。

cleaned\_dataset = terror.as\_matrix()

cleaned\_dataset = cleaned\_dataset[cleaned\_dataset[:,0].argsort()]

print(type(cleaned\_dataset))

row = len(cleaned\_dataset)

cl = len(cleaned\_dataset[0])

for idx, i in enumerate(cleaned\_dataset):

cleaned\_dataset[idx][0] = idx

print(cleaned\_dataset[-1])

print(len(cleaned\_dataset))

# 新建一列 test ： 1表示属于test集， 0 表示属于train

append\_ar = np.zeros((row,1), dtype='int64')

for idx, r in enumerate(cleaned\_dataset):

# 先拿出2015-2016

# 再拿出unknown

if (r[2] == 2015) or (r[2] == 2016):

#gname = unkown =>>'0000000001'

if r[-1] == 1.0:

append\_ar[idx] = 1 # 是测试样本

else:

append\_ar[idx] = 0 # 2015-2016 间的已知gname，是训练样本

elif r[-1] == 1.0:

append\_ar[idx] = 11 # 其他年份的unknown类型， 可能是训练样本，也可舍弃。

else:

append\_ar[idx] = 12 # 其他年份的已知gname ,是训练样本。

# 在最后一列添加test标签

append\_dataset = np.append(cleaned\_dataset, append\_ar,axis =1)

# 提取测试样本

test = np.array( [r[:-1] for r in append\_dataset if r[-1] == 1 ])

num\_test = len(test)

print(test[-10])

# 提取训练样本

train = np.array( [r[:-1] for r in append\_dataset if r[-1] != 1])

num\_train = len(train)

print('feature dim %s' % len(train[0]))

print('train num is %s'% num\_train)

print('test num is %s' % num\_test)

print('total sum is %s '% row)

print(num\_test + num\_train)

# 提取不包含 ‘unknow’ 类型的训练样本

train\_no\_unknown = np.array( [r[:-1] for r in append\_dataset if (r[-1] == 0 or r[-1] == 12)])

num\_train\_no\_unknown = len(train\_no\_unknown)

# to generate groups

# row: group\_id, line: event\_id

def gen\_groups(dataset):

# dataset is a feature matrix , for each row, row[1] is eventid, row[0] is gname

# return sorted gname, row[0] is gname, row[1] is eventid

# return group\_ids

print(dataset[:10])

groups = dataset[dataset[:,0].argsort()]

ngroup = []

group\_1 = [row[1] for row in groups if row[0] == 1]

print(group\_1)

if int(len(group\_1)) != 0:

ngroup.append(group\_1)

#print(group\_1)

group\_id = 2

temp = []

tag = int(len(group\_1))

print(tag)

for row in groups[tag:]:

#print(row[0])

if row[0] == group\_id:

temp.append(row[1])

#print(temp)

elif len(temp) != 1:

#print(temp)

ngroup.append(temp)

group\_id = group\_id + 1

temp = [row[1]]

else:

#print('temp empty when group\_id = %d' % group\_id)

group\_id = group\_id + 1

temp = []

print('final goup\_id is %d' % group\_id)

return ngroup

train\_groups = gen\_groups(train\_no\_unknown[:,[-1,0]])

print(train\_groups[0])

print(len(train\_groups))

# generate train dataset with pos\_pairs, neg\_pairs and labels

# asume we have got group\_id and member list, we have a 2d list Groups, index is the groupid, row is all the eventid.

# for each group, generate negtive samples number

neg\_count = 5

def gen\_pairs(Groups, idx, neg\_count):

# to generate positive and negtive samples for the idx group

# neg\_percent: the number of negtive samples to generate = total negetive samples \* neg\_percent

pos\_pairs = [[x, y ] for idx\_x, x in enumerate(Groups[idx]) for idx\_y, y in enumerate(Groups[idx]) if idx\_x != idx\_y]

num\_pos = len(pos\_pairs)

pos\_labels = np.ones(num\_pos)

# randomly generate negtive samples

G = Groups

np.delete(G, (idx), axis=0)

neg\_pairs = []

flat\_G = [event for g in G for event in g]

for event in Groups[idx]:

np.random.shuffle(flat\_G)

neg\_data = flat\_G[:neg\_count]

neg\_pairs = neg\_pairs + [[event, neg\_event] for neg\_event in neg\_data]

neg\_labels = np.ones(len(neg\_pairs))

pairs = pos\_pairs + neg\_pairs

labels = [p for p in pos\_labels] + [n for n in neg\_labels]

return pairs, labels

def gen\_all\_pairs(Groups):

data = [gen\_pairs(Groups, idx, neg\_count) for idx,g in enumerate(Groups)]

Data\_pairs, Labels = zip(\*data)

return Data\_pairs, Labels

def gen\_train(Groups):

data = [gen\_pairs(Groups, idx, neg\_count) for idx,g in enumerate(Groups) if idx < 10]

print(data[:2])

train\_pairs, y\_train0 = zip(\*data)

print(len(train\_pairs))

print(len(train\_pairs[0]))

print(len(y\_train0))

print(train\_pairs[:2])

print(y\_train0[:2])

# concate the feature vector of two samples as one convated feature vector

X\_train1 = [cleaned\_dataset[int(pair[0])] + cleaned\_dataset[int(pair[1])] for pairs in train\_pairs for pair in pairs]

print('number of x\_train1 is %s' % len(X\_train1))

# 正反调换来一遍

X\_train2 = [np.append(pair[1], pair[0],axis =1) for pair in train\_pairs]

# 两个特征差分试一下

X\_train = X\_train1 #+ X\_train2

y\_train = [y for y in y\_train0 for yy in y] #+ y\_train0

return X\_train, y\_train

X\_train, y\_train = gen\_train(train\_groups)

# generate test dataset

def gen\_test\_pairs(fdata):

# fdata is the feature matrix of test data

# return concated feature, event\_id pairs

X\_test = []

test\_ids = []

for idx, event in enumerate(fdata):

test\_id = [[event[0],e[0]] for e in fdata[idx+1:]]

test\_ids = test\_ids + test\_id

test\_feature = [np.append(event, e, axis = 1) for e in fdata[idx+1:]]

X\_test = X\_test + test\_feature

return X\_test, test\_ids

X\_test, y\_test = gen\_train(test)

print('nnum of test is %d' % len(y\_test))

# xgboost//classifier

print(X\_train[:2])

print(y\_train[:2])

print(type(X\_train))

# fit model no training data

model = XGBClassifier()

model.fit(X\_train, y\_train)

print(model)

# make predictions for test data

y\_pred = model.predict(X\_test)

'''

predictions = [round(value) for value in y\_pred]

print(y\_pred[:100])

# evaluate predictions

accuracy = accuracy\_score(y\_test, predictions)

print("Accuracy: %.2f%%" % (accuracy \* 100.0))

'''

# grouping the y\_pred

import matplotlib.pyplot as plt

%matplotlib inline

G = nx.read\_edgelist('test\_g.edgelist', nodetype=int, create\_using=nx.DiGraph())

G = G.to\_undirected()

nx.draw(G)

g = nx.karate\_club\_graph()

fig, ax = plt.subplots(1, 1, figsize=(8, 6));

nx.draw\_networkx(g, ax=ax)