2020 年第五届“数维杯”大学生
数学建模竞赛论文

关于网络评论情感倾向分析的合理性研究

摘　要

随着网络世界的不断发展，公共危机事件爆发时，信息在极短时间内迅速传播开来，引起群众的广泛关注。针对问题 1，采用感情色彩分析法，用 Python 脚本语言进行相关分值计算；针对问题 2，采用 Python 爬虫方法解决；针对问题 3，采用 BP 神经网络预测方法解决；针对问题 4，采用层次分析法解决。

对于问题 1 我们首先建立了关键字爬取模型。首先选取所需主题；其次将所选主题的关键字、词、句、段进行整合；再利用 Python 的第三方库 jieba 库对其进行分词，和相关的感情色彩分析进行数据爬取，使与主题有关的评论设为“1”，无关为“0”，将“1”全部整合，完成筛选任务。并借助 0-1 整数规划法和 Pycharm 软件得出关键字爬取的筛选方法。

对于问题 2 我们首先建立了层次分析模型。首先查找舆情评论数据，记录所有评论的发表时间、关注人数、评论人数及具体内容价值；再将四个因素分别以 0.1997、0.3343、0.2167、0.2493 为指标权重，计算每个评价的最终得分；最后将所有评价得分从高到低依次排列，抓取得分较高的评论。并借助层次分析法和 MATLAB 软件得出抓取方法。

对于问题 3 我们首先建立了 BP 神经网络模型。首先利用 Python 爬虫对所有评论数据进行关键字爬取，即通过“感情色彩”软件，确定感情倾向与程度；再通过适当降低负面评论在抓取时的最终分数，增加正面评论的最终分数得出抓取的最终分数，完成干预。并借助 BP 神经网络预测和 MATLAB 得出干预方法。

对于问题 4 我们首先建立了层次分析模型。首先查找舆情数据，记录全部舆情的传播时间、规模及网民情感倾向；再将传播时间、规模及网民情感倾向分别以 0.44、0.22、0.34 为指标权重，计算每个舆情的最终得分；最后依据得分，分为 1、2、3 等级，实行不同等级的干预方法。并借助层次分析法和 MATLAB 软件得出处理等级的划分方法。

在研究时间允许的情况下，针对关键字爬取模型进行适当的修改与优化，将所用的关键点进行无主观性的重新选取，提高得出结论的科学性，并可以在教育、绿植规划、网警巡检等方面进行模型推广。

关键词： 感情色彩分析，层次分析法，BP 神经网络模型，Python 爬虫

一、问题重述

1.1 针对问题 1：

随着网络中信息的迅速传播，网民评论的数量剧增，而负面报道或主观片面的一些失实评判常在一定程度上激发人们的危机感，甚至影响到政府及公共单位的公信力，影响到企业的形象及口碑。而舆情的情感倾向分析的首要任务是从繁多的评论中筛选出针对某一主题的舆情评论。需要解决的问题：

（1）从企业、政府角度出发，在所有主题中确定一个寻找主题；

（2）从市民、企业、政府角度出发，在所有主题评论中找出与此主题相关的所有评论，从而得出针对某一主题的舆情筛选方法。

1.2 针对问题 2：

网络信息的迅速传播，在网民的主观评断后，常影响到社会风气与企业、政府的形象。于是相关数据的抓取显得尤为重要，影响抓取的因素也有许多，例如：发表时间、关注人数、评论人数及具体内容价值等等。需要解决的问题：

（1）在筛选后所确定的评论范围内，通过数据整理，总结出所有数据的发表时间、关注人数、评论人数及具体内容价值；

（2）对数据的发表时间、关注人数、评论人数及具体内容价值进行总体分析，得出适合的全新数据抓取方法。

1.3 针对问题 3：

不同的舆情对不同的人群存在着不同的价值，不同的人员在舆情传播过程中起到了不同的作用。对于舆情的处理方式也就成为了一大难题，稍有不慎舆情风波将会掀起更高的巨浪。因此对网民们情感倾向进行引导，逐步转向对政府或企业有利的方向成为工作之重。需要解决的问题：

（1）预测舆情的情感倾向和发展趋势；

（2）对所有评论和报道进行感情色彩分析，并以是否对政府或企业有利为标准进行正负面感情程度分类。

（3）对所有评论进行人为干预，从而使感情倾向与趋势逐步转向对政府或企业有利的方向，并得出所需的干预方法。

1.4 针对问题 4：

不同舆情的传播速度具有一定的差异，管理部门检测到的舆情时间点并不固定，对于政府或企业而言对处于不同阶段的舆情需要进行干预的等级不同，划分等级问题成为了一大难题。需要解决的问题：

（1）统计舆情评论的疫情传播时间、规模及网民情感倾向；

（2）确定三个因素对舆情需要进行干预的等级的影响程度大小；

（3）结合舆情的评论数据，对舆情进行等级划分。从而得出舆情处理等级的划分方法。

二、问题分析

2.1 对于问题 1 的分析

基于重述中的两个问题，问题 1 所要研究的即为数据的筛选法方法，好的筛选方法可以更好地让企业、政府得知网民对于舆情的观点与态度，从而更以进行后续工作，避免无关数据的影响，节约时间、提高效率。

问题 1 属于归类判别问题，对于此类问题应利用统计学对于所给数据进行统计，并规定一判断标准完成分类任务。对于题目所给附件数据可以得知，众多的网络评论数据涉及到社会生活中的方方面面，因此应确定一企业、政府相关程度较大的单一主题，对应此主题，对所有评论进行符合判断标准的筛选。问题 1 所要结果为一筛选方式，因此针对特定主题的筛选思路与应用的数学方法是本题的关键所在。

由于以上原因，我们首先可以建立一个关于附件关键字、词、句、段的关键字爬取模型，然后将附录中的评论数据放入模型，利用 0-1 整数规划法去确定某一评论是否关于这一主题。

2.2 对于问题 2 的分析

基于问题重述中的问题，问题 2 所研究的是数据的随机抓取方法，抓取方法可以更好的让阅读者了解此舆情的发展情况，并自我预测其发展方向，也可以使企业、政府更好地做出应对方案。

问题 2 属于最优化问题，对于此类问题可以利用多因素综合分析解方程组的方法完成。对于本题，评论的抓取已经在完成与主题相关评论筛选的背景下进行，而影响此类抓取的因素众多，例如发表时间、评论人数、关注人数及具体内容等。因此，在忽略评论内容长短、切题的程度、评论的引申和可研究度问题的情况下，对发表时间、评论人数、关注人数及具体内容进行综合分析。问题 2 所求结果为一种新型抓取方法，所以如何去取得综合性较为优秀的评论为本题重点。

由于以上原因，我们首先可以建立层次分析模型，然后将已经筛选好的评论数据按发表时间、评论人数、关注人数及具体内容四个因素进行综合分析，利用层次分析决定抓取的评论。

2.3 对于问题 3 的分析

针对上述三个问题，舆情的处理方式一旦不符合大众的意向，便会导致舆情的另一种传播途径的产生与更负面的影响，对于问题 3 的探究就可以较好地解决此类问题，使舆情更容易转向对企业和政府有利的方向，更好地度过舆情。

问题 3 属于预测模块类问题，可以利用 BP 神经网络进行舆情的发展预测和实行方法后的发展预测。对于本题，应对评论进行情感分析，做出其正、负面程度的判定，然后进行发展趋势的预测。问题 3 所求为一种合理的干预方式，因此预测的结果以及选区的干预方式为本题的关键。

由于以上原因，我们首先建立关键字爬取模型，利用感情色彩分析对评论进行正、负面程度的判定，再利用 BP 神经网络预测进行干预方式的选取。

2.4 对于问题 4 的分析

针对上述三个问题，问题 4 研究的是针对舆情的处理等级的划分方法，好的等级划分方法可以将等级较高的舆情进行及时的干预，并且避免对等级较低的舆情进行无必要的干预，节约了人力、物力、财力。

问题 4 属于层次分析问题，对于此类问题可以利用层次分析将舆情传播时间、规模及网民情感倾向作为判定的因素进行综合分析。对于本题，在干预措施已经完备的情况下，由于可以影响划分等级的因素较多，在忽略其他条件下，以舆情传播时间、规模及网民情感倾向为判定标准，进行综合分析。问题 4 所求为一种划分方法，如何判定某一个舆情的等级成为了本题关键。

由于以上原因，我们可以首先建立层次分析模型，以舆情传播时间、规模及网民情感倾向为判定标准，再计算等级，得出最终的等级划分方法。

三、模型假设

1. 假设题目所给信息真实可靠；
2. 忽略除了所考虑因素以外的对所求方法有影响的因素；
3. 对于情感分析问题，将情感词的表达程度以定性问题转化为定量问题，关键字、词、句、段的分数标准大小充斥一定的主观意见，假设每个关键字的程度均相同，词、句、段也即是此假设；
4. 假设个人所整理使用的评论数据足够大，可忽略偶然性的发生。

四、定义与符号说明

符号定义	符号说明
A	矩阵
CI	一致性指标
CR	一致性比例
MSE	均方误差
Epoch	所有训练样本的一个正向传递和一个反向传递。
Epochs	所有训练样本的多个正向传递和反向传递。

图 4

五、模型的建立与求解

数据的预处理：

1. 将过长评论或在一条评论中重复多次的相同评论语言进行删除、整理；
2. 在所给附件中选取部分关键字、词、句、段，使选取的关键字、词、句、段与“大学排名的定制内容”主题相关程度较大；
3. 在抖音、新浪微博等软件选取关于疫情的网民评论和热点，统计、整理其评论量、关注量、点赞量、发表时间、具体内容价值等信息，并且评论随机取样，共采用了 500 组评论与 180 组相关舆情（具体信息请见附录）；
4. 对问题 4 中的信息，将舆情的传播规模以评论量代替；传播时间以舆情的发表时间至统计时的时间间隔来代替；网民情感倾向以点赞量的数量来代替。

5.1 问题 1 的模型建立与求解

5.1.1 关键字爬取模型的建立

通过对问题的分析与假设，我们已经对问题的关键与筛选思路有一定的认识。我们需要解决的问题是如何得出一种针对某一主题的舆情筛选方法，题目的要求是结合附件 1 中给出的数据进行方法确定，剔除重复与不相关的评论数据后选用关键字爬取模型进行分析。具体步骤如下：

（1）将题目所给附件中关于“大学排名的定制内容”这一主题所涉及到的关键字、词、句、段进行大致总结，例如“某某大学”、“高考成绩要求”、“男女比例”、“学习风气”等；

（2）将关键字、词、句、段为寻找标准，寻找全部数据内包含此类标准的舆情评论，并将符合标准的评论设为“1”，不符合标准的评论设为“0”；

（3）将所有的“1”进行整合，完成筛选。

（4）具体见下程序框图：

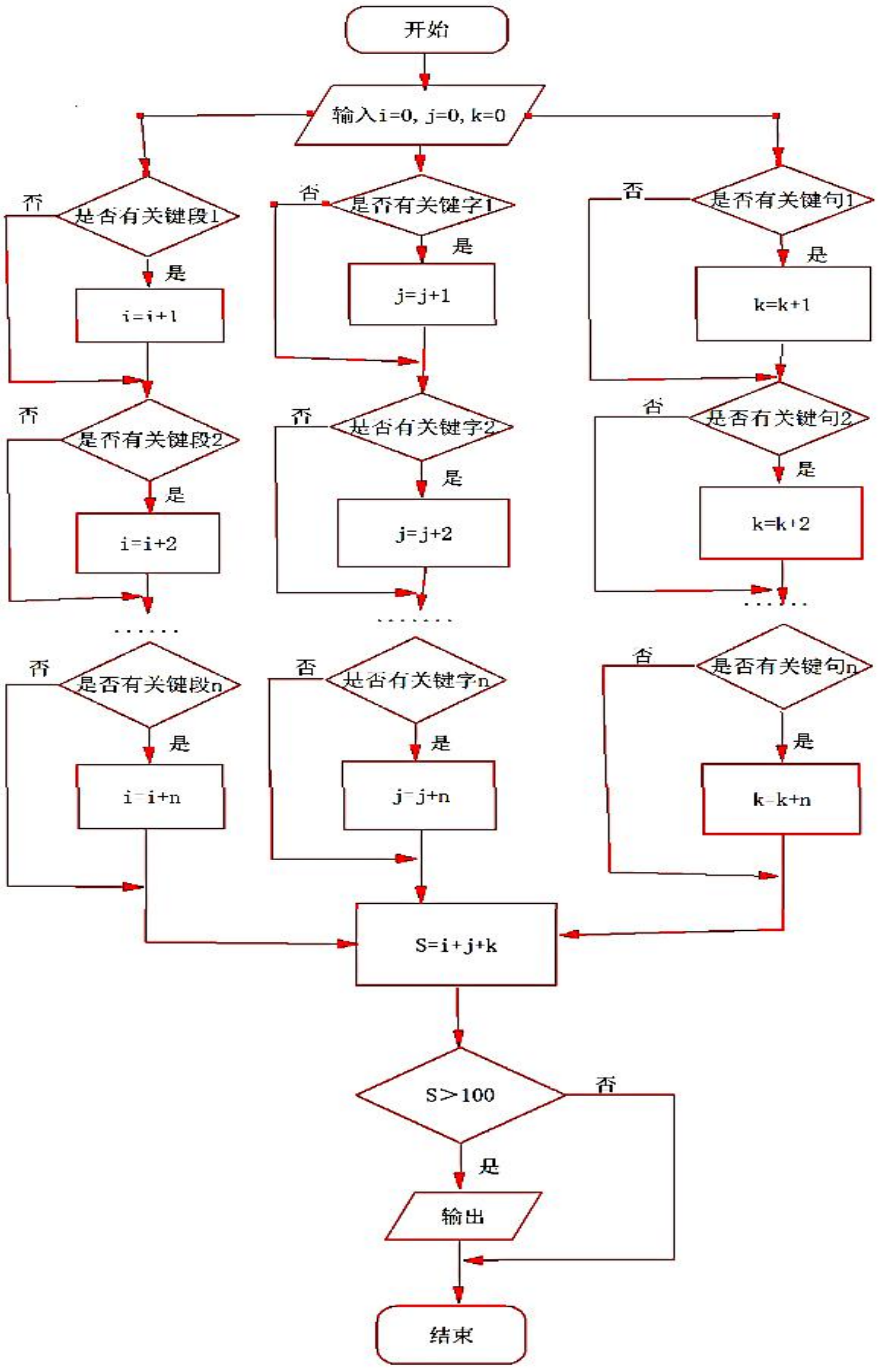

图 5.1.1

5.1.2 关键字爬取模型的求解

将预处理数据带入上述模型中，通过 Pycharm 软件得到全部的筛选结果（编程代码详见附录）。

筛选方法为：

①选取所需主题；

②将所选主题的关键字、词、句、段进行整合；

③利用 Python 爬虫进行数据爬取，使与主题有关的评论设为“1”，无关为“0”，将“1”全部整合，完成筛选任务。

5.1.3 结果

经过上述过程分析与实验，此类关键字爬取的筛选方式可以进行实际应用。即：

①选取所需主题；

②将所选主题的关键字、词、句、段进行整合；

③利用 Python 爬虫进行数据爬取，使与主题有关的评论设为“1”，无关为“0”，将“1”全部整合，完成筛选任务；

利用此方法，可以较为快速地完成大数据舆情评论的筛选任务，从而使得企业、政府对舆情后续干预工作进行更加迅速。

5.2 问题 2 的模型建立与求解

5.2.1 层次分析模型的建立

通过对问题 2 的分析与假设，可以得知我们需要解决的问题重点是如何去评定舆情评价的重要性和价值度。题目的要求是在确定影响因素的前提下，对各个影响因素进行其对舆情评论价值的影响，并且对各个评价进行综合性分析。剔除部分具有偶然性的数据后，选用层次分析模型进行分析。具体步骤如下：

（1）查找数据，随机选取部分评论并记录其发表时间、关注人数、评论人数及具体内容价值，并且尽可能使数据足够大，减少数据偶然性的影响；

（2）将发表时间、关注人数、评论人数及具体内容价值作为评判标准，利用一致性指标运算预算处四个因素的指标权重；

首先，利用下列五个标准来表示因素的重要程度：

因素重要程度标准表	
标度	含义
1	两个因素相比，同样重要
2	两个因素相比，一个因素比另一个因素稍微重要
3	两个因素相比，一个因素比另一个 因素重要
4	两个因素相比，一个因素比另一个 因素明显重要
5	两个因素相比，一个因素比另一个 因素重要得太多了

图 5.2.1-1

①我们先衡量以下四个指标的关联度，相关性，重要性，如下表：

发表时间、评论人数、关注人数、价值重要程度表

	发表时间	评论人数	关注人数	价值
发表时间	1	2/3	4/3	1/2
评论人数	3/2	1	2	4/3
关注人数	3/4	1/2	1	3/2
价值	2	3/4	2/3	1

图 5.2.1-2

通过输入矩阵 A=[1 2/3 4/3 1/2;3/2 1 2 4/3;3/4 1/2 1 3/2;2 3/4 2/3 1]
特征值法求权重的结果为：

0.1997
0.3343
0.2167
0.2493

得出一致性指标为：

CI=0.0624

得出一致性比例为：

CR=0.0702

因为 CR<0.10，所以该判断矩阵 A 的一致性可以接受。

②在发表时间的影响下，选取随即三个评论，以评论 1、2、3 作为指标，建立下图：

发表时间与评论 1、2、3 指标图

发表时间	评论 1	评论 2	评论 3
评论 1	1	1/2	3
评论 2	2	1	5
评论 3	1/3	1/5	1

图 5.2.1-3

通过输入矩阵 A=[1 1/2 3;2 1 5;1/3 1/5 1]
特征值法求权重的结果为：

0.3090
0.5816
0.1095

得出一致性指标为：

CI=0.0018

得出一致性比例为：

CR=0.0036

因为 CR<0.10，所以该判断矩阵 A 的一致性可以接受。

③在关注人数的影响下，以评论 1、2、3 作为指标，建立下图：

关注人数与评论 1、2、3 指标图

关注人数	评论 1	评论 2	评论 3
评论 1	1	1/5	2
评论 2	5	1	8
评论 3	1/2	1/8	1

图 5.2.1-4

通过输入矩阵 A=[1 1/5 2;5 1 8;1/2 1/8 1]
特征值法求权重的结果为：

0.1618
0.7510
0.0872

得出一致性指标为：

CI=0.0028

得出一致性比例为：

CR=0.0053

因为 CR<0.10，所以该判断矩阵 A 的一致性可以接受。

④在关注人数的影响下，以评论 1、2、3 作为指标，建立下图：

评论人数与评论 1、2、3 指标图

评论人数	评论 1	评论 2	评论 3
评论 1	1	1/4	1/7
评论 2	4	1	1/2
评论 3	7	2	1

图 5.2.1-5

通过输入矩阵 A=[1 1/4 1/7;4 1 1/2;7 2 1]
特征值法求权重的结果为：

0.0823
0.3150
0.6026

得出一致性指标为：

CI=9.9075e-04

得出一致性比例为：

CR=0.0019

因为 CR<0.10，所以该判断矩阵 A 的一致性可以接受。

⑤在评论价值的影响下，以评论 1、2、3 作为指标，建立下图：

评论价值与评论 1、2、3 指标图

价值	评论 1	评论 2	评论 3
评论 1	1	1/4	1/5
评论 2	4	1	2/3
评论 3	5	3/2	1

图 5.2.1-6

通过输入矩阵 A=[1 1/4 1/5;4 1 2/3;5 3/2 1]
特征值法求权重的结果为：

0.0992
0.3735
0.5272

得出一致性指标为：

CI=0.0018

得出一致性比例为：

CR=0.0036

因为 CR<0.10，所以该判断矩阵 A 的一致性可以接受。

（3）再以指标权重乘以每个评价相对应的指标等级，得到评价的最终得分，以最终得分；

（4）比较每个评价的最终得分，得出抓取的标准。

5.2.2 层次分析模型的求解

将预处理的数据带入上述模型中，通过 MATLAB 软件用一致性指标运算出每个影响因素的指标权重，并计算出每条舆情评论的最终得分，依据得分的高低抓取所需的评论（编程代码详见附录）。

最后通过归一化，我们进行数据的汇总分析：

发表时间、评论人数、关注人数、评论价值与评论 1、2、3 权重指标图

	权重指标	评论 1	评论 2	评论 3
发表时间	0.1997	0.3090	0.5816	0.1095
评论人数	0.3343	0.0823	0.3150	0.6026
关注人数	0.2167	0.1618	0.7510	0.0872
价值	0.2493	0.0992	0.3735	0.5272

图 5.2.2

得出最终得分：评论 1 得分: 0.15
评论 2 得分: 0.47
评论 3 得分: 0.38

因此三条评论相较之下，评论 2 最应先被抓取，其次评论 3，再次之评论 1。

得出抓取方法为：

①查找舆情评论数据，在筛选完成的背景下，记录所有评论的发表时间、关注人数、评论人数及具体内容价值；

②将发表时间、关注人数、评论人数及具体内容价值分别以 0.1997、0.3343、0.2167、0.2493 为指标权重，计算每个评价的最终得分；

③将所有评价得分从高到低依次排列，抓取得分较高的评论。

5.2.3 结果

经过上述的实验分析与进行，此类利用层次分析模型得出的抓取方法可行。即：

①查找舆情评论数据，在筛选完成的背景下，记录所有评论的发表时间、关注人数、评论人数及具体内容价值；

②将发表时间、关注人数、评论人数及具体内容价值分别按 0.1997、0.3343、0.2167、0.2493 为指标权重，计算每个评价的最终得分；

③将所有评价得分从高到低依次排列，抓取得分较高的评论。

利用此抓取方法，可以更好的让企业、政府了解此舆情的发展情况，并自我预测其发展方向，也可以使企业、政府更好地做出应对方案，避免更严重的损失，完成自我干预。

5.3 问题 3 的模型建立与求解

5.3.1 BP 神经网络模型的建立

通过对问题 3 的分析与假设得知，我们需要解决的问题是：影响网民情感倾向和趋势的影响程度众多，而情感的分析更多的是一种定型的分析，因此需先将定量分析引入感情程度的划分，再将舆情评论的情感程度带入预测，决定所需要的干预方法。在剔除问题 2 中部分无意义的评论数据后，进行感情分析，并预测干预前后的网民感情倾向。

具体步骤如下：

（1）将已经抓取好的评论数据内容进行情绪关键字、词、句、段抓取，用 python 的第三方库 jieba 库进行文本情感分析，并通过合理的算法对其进行评分，以确定评论的情感倾向与程度评分；

其中具体思路与流程，见下程序框图：

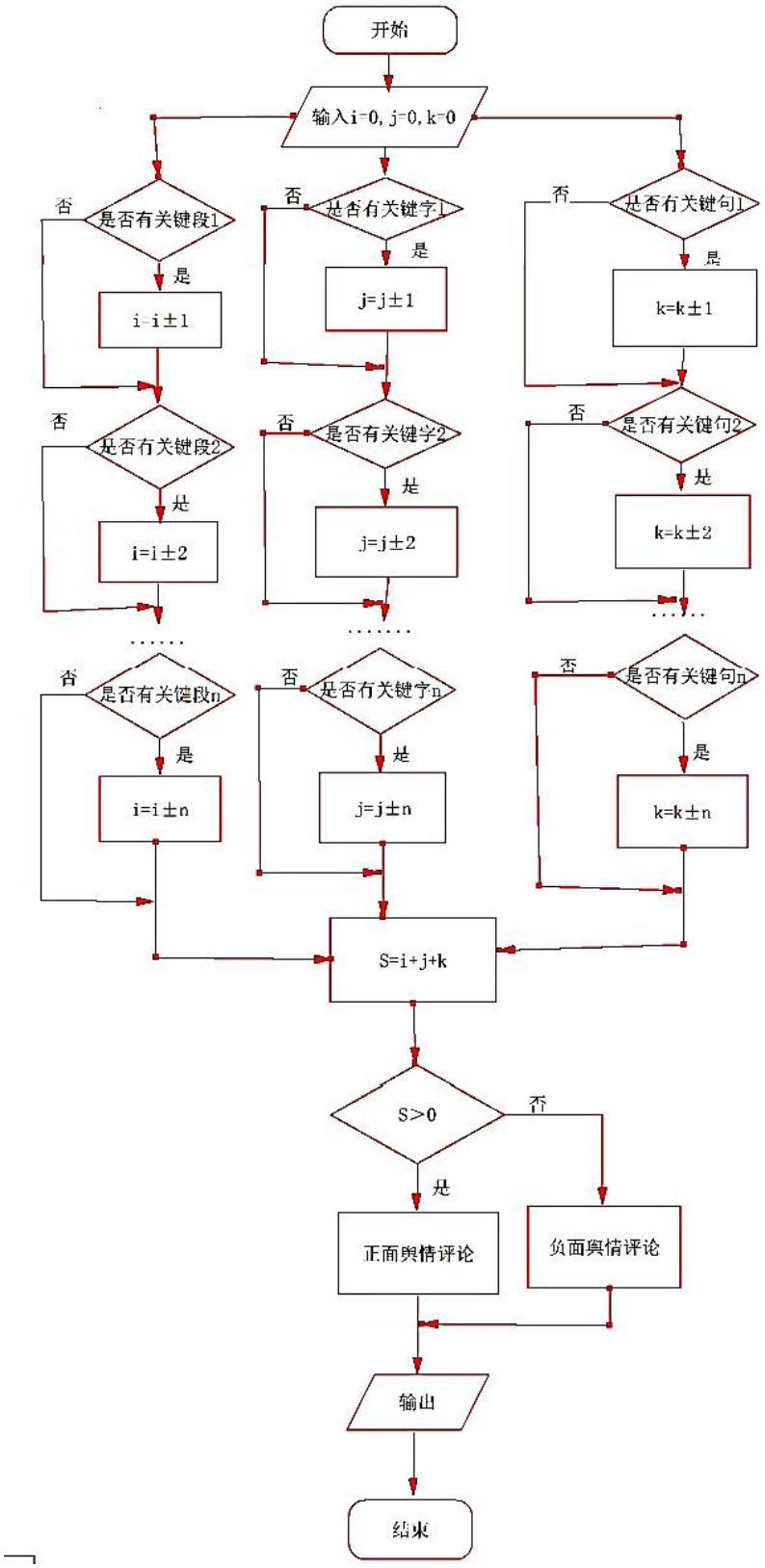

图 5.3.1

（2）将已经划分好程度的评论进行 BP 神经网络预测，来确定在干预前网民的网络倾向与趋势；

通过传播时长，规模，网民情感，来衡量一个舆论，热点的价值，我们用（1,4）来量化价值，越接近于 4 说明其价值量越大，越接近于 1，表明其越无价值。我们用 160 组数据来作为神经网络模型的训练组进行训练，综合三种训练方法，考虑到莱文贝格-马夸特方法能提供数非线性最小化 （局部最小）的数值解，选择了其中比较快的莱文贝格-马夸特方法进行训练。

其中共训练了 15Epochs，其中在第 Epoc9 的时候 MSE=0.14778 达到最小误差，其具体训练结果如下图所示：

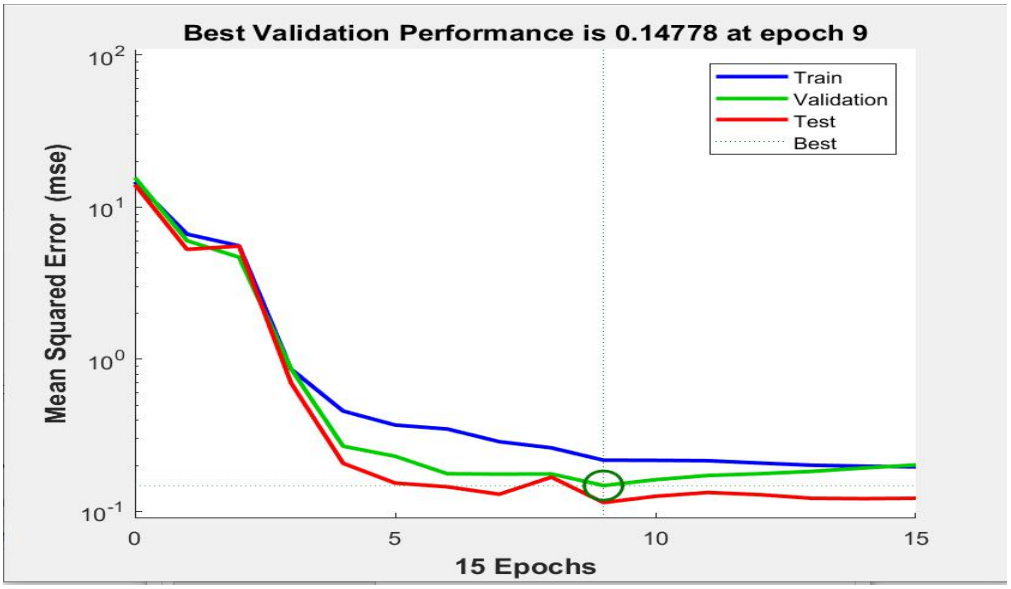

图 5.3.1-1

其回归分析图如下：

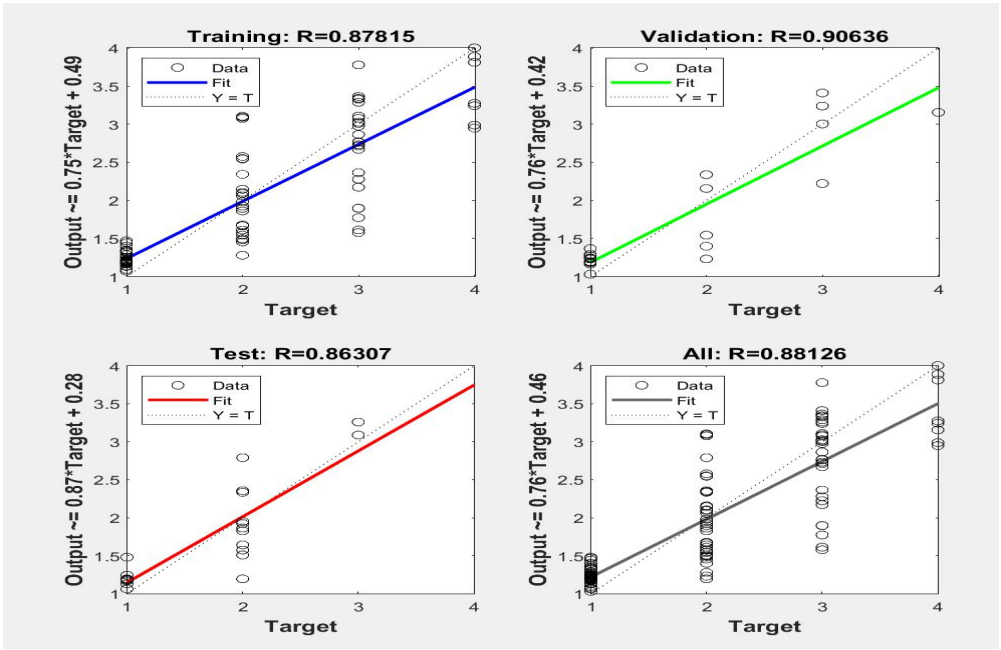

图 5.3.1-2

由此可见 Training,Validation,Test 组的 R 值都接近于 1，然后综合 All 的结果，R=0.88126，结果相对比较准确。

（3）进行企业、政府干预，再对干预后的情况进行 BP 神经网络预测，通过将干预前后的两种情况进行比较，得出干预方式。

5.3.2 BP 神经网络模型的求解

将预处理的数据带入上述模型中，通过“感情色彩”软件，即关键字爬取模型确定情感倾向，并通过人为干预，减少对政府或企业不利的评论数目，并将部分负面评论在抓取时分数降低，将正面评论的抓取分数升高，完成干预。

预处理的数据带入后，进行神经网络预测，得出共 20 组预测值（预测代码见附录），全部预测值的拟合图如下：

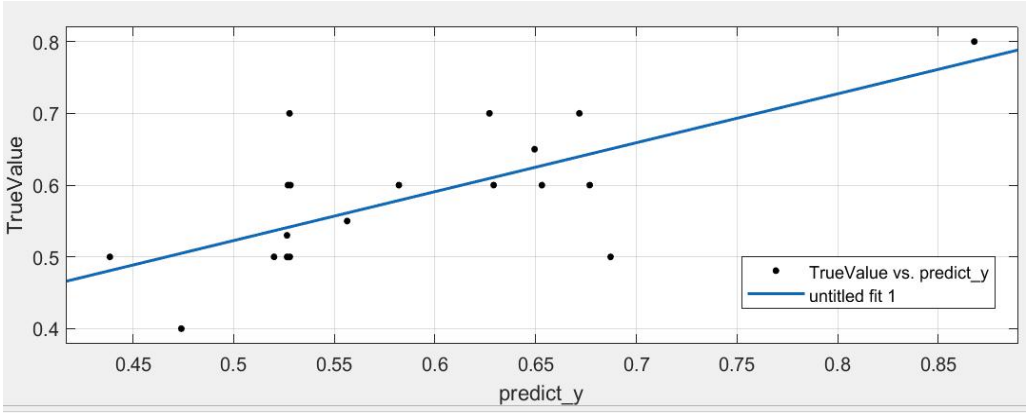

图 5.3.2

通过神经网络的一个对人们情感的一个大致预测，以便政府机关，社会做好相应的准备，来应对这种舆论走向，以正确的引导人们的情感变化。

于是得出干预方法如下：

①将所有评论通过“感情色彩”软件，确定感情倾向与程度；

②通过适当降低负面评论在抓取时的最终分数，适当增加正面评论的最终分数得出抓取的最终分数，完成干预。

5.3.3 结果

经过上述实验的进行与研究，得出此种干预方法可行。即：

①将所有评论通过“感情色彩”软件，确定感情倾向与程度；

②通过适当降低负面评论在抓取时的最终分数，适当增加正面评论的最终分数得出抓取的最终分数，完成干预。

利用此方法，可以在避免舆情评论在一并删除时激起网民的深一步风波的同时，在一定程度上使舆情情况逐步转向对政府或企业有利的方向，多次干预，程度多次加深。

5.4 问题 4 的模型建立与求解

5.4.1 层次分析模型的建立

通过对问题 4 的分析与假设，我们需要解决的问题是：统计舆情评论的疫情传播时间、规模及网民情感倾向；确定三个因素对舆情需要进行干预的等级的影响程度大小；以及结合舆情的评论数据，对舆情进行等级划分。从而得出一种舆情处理等级的划分方法。具体步骤如下：

（1）查找一部分舆情数据，统计所有舆情的传播时间、规模及网民情感倾向。将传播时间、规模及网民情感倾向通过一致性检验预算出其权重等级。

其相应的传播时间，传播规模，网民情感倾向的相关指标权重经大数据调研得出，在这我们将干预分为三个等级，级别越高，则干预越强。为了达到这个目标，我们评价的指标有疫情传播时间、规模以及网民情感倾向三个指标。因此我们采取分而治之的思想，两个两个指标进行比较，最终根据两两比较的结果来推算出权重。

首先，我们利用下列五个标准来表示因素的重要程度：

因素重要程度标准表	
标度	含义
1	两个因素相比，同样重要
2	两个因素相比，一个因素比另一个因素稍微重要
3	两个因素相比，一个因素比另一个 因素重要
4	两个因素相比，一个因素比另一个 因素明显重要
5	两个因素相比，一个因素比另一个 因素重要得太多了

图 5.4.1-1

其次，根据大量数据得调研，得出如下判断矩阵：

传播时间、规模、情感倾向重要程度表

	传播时间	传播规模	情感倾向
传播时间	1	1/2	2/3
传播规模	2	1	4/3
情感倾向	2/3	3/4	1

图 5.4.1-2

上表为一个 3*3 方阵，记为 A。

但是目前并不确定这个数值指标是否有很大的可信度，因此需要利用 Matlab 进行一致性检验。

一致性检验的分析如下：

①以传播时间、规模、网民情感倾向为指标得出表**图 5.4.1-2**，将一致性检验代码输入矩阵(代码见附录），得出特征值法求权重结果为：

0.2222
0.4444
0.3333

得出一致性指标为：

CI=-6.6613e-16

得出一致性比例为：

CR=-1.2810e-15

其中我们需要查询的平均随机一致性指标 RI 如下表：

平均随机一致性指标 RI 表									
n	1	2	3	4	5	6	7	8	9
RI	0	0	0.52	0.89	1.12	1.26	1.36	1.41	1.46

图 5.4.1-3

因为 CR<0.10，所以判断矩阵 A 的一致性可以接受。

②在传播规模的影响下，以干预等级 1、2、3 作为指标，建立下表：

舆情传播规模与干预等级 1、2、3 指标表			
传播规模	干预等级 1	干预等级 2	干预等级 3
干预等级 1	1	1/4	1/2
干预等级 2	4	1	3/2
干预等级 3	2	2/3	1

图 5.4.1-4

通过将一致性检验代码输入矩阵(代码见附录）得出特征值法求权重结果为：

0.1463
0.5317
0.3220

得出一致性指标为：

CI=0.0046

得出一致性比例为：

CR=0.0088

因为 CR<0.10，所以该判断矩阵 A 的一致性可以接受。

③在传播时间的影响下，以干预等级 1、2、3 作为指标，建立下表：

舆情传播时间与干预等级 1、2、3 指标表

传播时间	干预等级 1	干预等级 2	干预等级 3
干预等级 1	1	3/2	3
干预等级 2	2/3	1	2
干预等级 3	1/3	1/2	1

图 5.4.1-5

通过将一致性检验代码输入矩阵(代码见附录）得出特征值法求权重结果为：

0.5000
0.3333
0.1667

得出一致性指标为：

CI=-2.2204e-16

得出一致性比例为：

CR=-4.2701e-16

因为 CR<0.10，所以该判断矩阵 A 的一致性可以接受。

④在情感倾向的影响下，以干预等级 1、2、3 作为指标，建立下表：

网民情感倾向与干预等级 1、2、3 指标表

情感倾向	干预等级 1	干预等级 2	干预等级 3
干预等级 1	1	1/3	1/5
干预等级 2	3	1	2/3
干预等级 3	5	3/2	1

图 5.4.1-6

通过将一致性检验代码输入矩阵(代码见附录）得出特征值法求权重结果为：

0.1119
0.3478
0.5403

得出一致性指标为：

CI=6.1678e-04

得出一致性比例为：

CR=0.0012

因为 CR<0.10，所以该判断矩阵 A 的一致性可以接受。

（2）预算出某一舆情在传播时间、规模及网民情感倾向下分别所得的权重得分。

（3）计算本舆情的最终得分，依据得分，划分等级。

5.4.2 层次分析模型的求解

将预处理的数据带入上述层次分析模型，通过 MATLAB 软件用一致性指标预算出传播时间、规模及网民情感倾向的指标权重，并计算出每条舆情评论的最终得分，依据得分将舆情等级划分为 1、2、3 三个等级。

通过归一化，带入某一评论相应的权重指标分数，最后进行数据汇总与最终得分计算：

传播规模、时间、网民情感倾向与干预等级层次分析表

	权重指标	干预等级 1	干预等级 2	干预等级 3
传播规模	0.44	0.15	0.53	0.32
传播时间	0.22	0.5	0.33	0.17
情感倾向	0.34	0.11	0.35	0.54

图 5.4.2

干预等级 1 得分：0.2134；

干预等级 2 得分：0.4248；

干预等级 3 得分：0.3618；

因此我们对当前的疫情舆情的最佳干预等级为干预等级 2。

同理，对这类似的舆情，我们得首先做一个数据调研，然后根据计算出各个指标得权重，用层次分析法得出最佳干预方案。

划分方法如下：

①查找舆情数据，记录每一条舆情的传播时间、规模及网民情感倾向。

②将传播时间、规模及网民情感倾向分别以 0.44、0.22、0.34 为指标权重，计算每个舆情的最终得分。

③依据得分，将最终得分在为 1、2、3 等级，实行不同等级的干预方法。

5.4.3 结果

经过上述过程的分析，可知此种划分方法可行，即：

①查找舆情数据，记录每一条舆情的传播时间、规模及网民情感倾向。

②将传播时间、规模及网民情感倾向分别以 0.44、0.22、0.34 为指标权重，计算每个舆情的最终得分。

③依据得分，将最终得分在为 1、2、3 等级，实行不同等级的干预方法。

利用此方法，可以避免风波已过的舆情被再次干预，造成不必要的人力、物力、财力的浪费，也可以将当下舆情依照等级大小划分，更好的进行人为干预，快速止损，引导网民感情倾向与趋势，使其向对企业、政府有利方向进行。

六、模型的评价及优化

6.1 误差分析

6.1.1 针对于问题 1 的误差分析

问题 1 中数据主要为所给附件的筛选过程，又筛选时关键字、词、句、段的选取会产生人为主观认识的差异，因此可能造成了一些舆情评论的筛入或筛出。在程序框图中添加或减少一组或几组关键词，就会造成总评论数据库所筛选出的相关评论的数量变化。因此，问题 1 中的筛选方法，可以依据利用此方法的企业、政府等其在发展过程中接触较多的关键字、词、句、段进行有特色的筛选，可以将与自身关系较大的评论进行筛选。

6.1.2 针对于问题 2 的误差分析

问题 2 中误差容易出现在数据的选取中，本次解决问题共涉及到 500 组发表时间、评论人数、关注人数及具体内容的舆情评论，虽是随机选取，但因时间问题数据较少，无法避免偶然性的影响，减少的偶然性影响也较为有限。因此，在时间充足的实际社会生活中，可以在筛选舆情评论完成的背景条件下选取更多组数据，设置更多的影响因素，更好地减少偶然性的影响。

6.1.3 针对于问题 3 的误差分析

因问题 3 涉及到了问题 1 和问题 2 的模型建立方法，因此问题 3 拥有问题 1 与问题 2 分别存在的误差。感情分析中关键字、词、句、段的选取也存在人为主观因素的差异，可能造成了情感倾向程度的误差；而数据选取量的较少问题，也无法成功使得数据偶然性的发生降到最低。基于上述问题，可以使用现市场中的“感情色彩”软件，在情感分析较为成熟的分析软件中判定网民舆情评论的倾向和程度；同时，在时间允许下，选取更多数据进行分析预测，使得预测结果可以更为准确、更具有说服力。

6.1.4 针对于问题 4 的误差分析

因问题 4 中所使用的影响因素为舆情传播时间、规模及网民情感倾向，因时间原因，本题选用了 180 组数据进行探究，虽然 180 组均为随机取样，但受到的偶然性影响仍然较大，减弱的偶然性影响较为有限。因此将此方法运用到生活实际时，因受到的时间限制较少，可以选取更多组数据，并选取更多可以影响舆情等级的因素进行划分。

6.2 模型的优点

（1）四个问题的模型建立，均得到了针对舆情有效的应对方法，较好地解决了筛选、抓取、干预、划分方法的使用问题；

（2）利用程序框图使关键字、词、句、段的分析简化，使模型更简单易懂、使用原理显而易见；

（3）将情感分析这一定性分析转化为数值上的定量分析，使模型得到了简化，方法易于使用；

（4）利用层次分析，选取较为优质的评论，实用性强。

6.3 模型的缺点

（1）选用的数据总量较少，得出的结论受到偶然性影响较大；

（2）在关键字、词、句、段选取过程中，因人为主观性影响，选择出来的关键点主观性较强，而不同的人也有着不同的选取认知，差异性较大；

（3）因数据较少问题，得出的最终数值有一定偏差。

（4）层次分析模型中，通过一致性检验后得出的不同因素的指标权重数值并非准确值，是一个大概值。

6.4 模型的推广

6.4.1 针对关键字爬取模型

问题 1 以及问题 3 中的关键字爬取模型在数学建模类型中是一种新的思路，其对一系列特定数据的寻找和分类有着极大的便利，但是也存在一定的范围限制和主观性，对待某一特定数据的不同关键词理解就会导致得到的结果不同。但 app“感情分析”即是此种方法的一种系统性运用。因此证明，此种方法是可以推广至众多区域。

例如：

（1）在教育方面，可以作为作文评定时，是否切题时的一种标准；

（2）在环境保护方面，建筑绿植选取合适地域时范围规划问题；

（3）网警查阅违规评论，利用程序快速搜索不正当语言时，加快效率。

6.4.2 针对层次分析模型

问题 2 及问题 4 所涉及到的层次分析模型，是在数学建模过程中经常用到的算法模型，其对多因素、多标准、多方案的综合评价及趋势预测相当有效，并且可以将多因素问题进行综合评价，逐层分解变为多个单准则评价问题。但缺点也非常明显，因其需要进行一致性检验，因此一旦检验不成功，便失去了使用意义；其次还需要专家的数据支撑，若给出的指标不合理，所得结果自然也并不准确。但也因为他的多因素综合分析，推广意义更为重要：

（1）依据层次分析的综合得分，可以将最终得出的分数进行等级判定，综合体现某一研究对象的优劣。

（2）在日常生活中，可以利用层次分析去选择适合自己的商品、大学、科研成果的价值等。

6.4.3 针对 BP 神经网络模型

问题 3 涉及到的 BP 神经网络模型，BP 神经网络模型具有高度的自我学习力和自适应能力，其泛化能力和容错能力也是一流。但是其依赖数据的样本选择、预测能力与实际问题的矛盾也是存在的。因此如何把握好其学习、预测的度，成为了此类模型的关键难题。不过 BP 神经网络近几年的发展迅猛也是肉眼可见，其推广的范围更加广阔：

（1）人工智能的自我学习，自我感知与预测；

（2）对于农业生产中，对天气状况的预测，以判断农业劳动者的下一步行动，避免因雨雪天降低收成量。

参考文献

［1］刘金硕，李哲，叶馨，陈嘉敏，邓娟.文本情感倾向性分析方法：bfsmPMI-SVM［J］.武汉大学学报.2017.第 63 卷 第 3 期.

［2］徐小星.网络舆情的倾向性分析及应用研究［D］.电子科技大学.2015.

［3］姜启源，谢金星，叶俊. 数学建模（第五版）［M］.北京.高等教育出版社.2006.1 至 213 页.

［4］赵东方. 数学建模与计算［M］.北京.科学出版社.2007.1 至 37 页.

［5］王根，赵军.基于多重标记 CRF 的句子情感分析研究［R］.全国第九届计算语言学学术会议.大连.清华大学出版社.2007.

附　录

1.层次分析法

```
1）%层次分析法判别矩阵的一致性检验代码
disp('请输入判断矩阵 A')
A=input('A=');
[n,n] = size(A);
特征值法求权重% % % % % % % % % % % % %
[V,D] = eig(A);
Max_eig = max(max(D));
[r,c]=find(D == Max_eig , 1);
disp('特征值法求权重的结果为：');
disp( V(:,c) ./ sum(V(:,c)) )
% % % % % % % % % % % %计算一致性比例 CR % % % % % % % % % % % % %
CI = (Max_eig - n) / (n-1);
RI=[0 0 0.52 0.89 1.12 1.26 1.36 1.41 1.46 1.49 1.52 1.54 1.56 1.58 1.59];  %注意哦，这里的 RI 最多支持 n = 15
% 这里 n=2 时，一定是一致矩阵，所以 CI = 0，我们为了避免分母为 0，将这里的第二个元素改为了很接近 0 的正数
CR=CI/RI(n);
disp('一致性指标 CI=');disp(CI);
disp('一致性比例 CR=');disp(CR);
if CR<0.10
    disp('因为 CR<0.10，所以该判断矩阵 A 的一致性可以接受!');
else
    disp('注意：CR >= 0.10，因此该判断矩阵 A 需要进行修改!');
end
```

2 .Python 代码

1)大学信息筛选代码

```
import bs4
import requests
from bs4 import BeautifulSoup


def getNetText(url):#从网络上获取大学排名网页内容
    try:
        r = requests.get(url,timeout = 40)
        r.raise_for_status()
        r.encoding=r.apparent_encoding
        return r.text
    except:
        return ""
```

```
def List(ulist,html):#提取网页内容中信息到合适的数据结构
    soup=BeautifulSoup(html,'html.parser')
    for tr in soup.find('tbody').children:

        if isinstance(tr,bs4.element.Tag):
            tds = tr('td')#tr('td')相当于 tr.find_all('td')
            ulist.append([tds[0].string,tds[1].string,tds[3].string])


def printList(ulist,num):#利用数据结构展示并输出结果
    tplt="{0:^2}\t{1:{3}^10}\t{2:^5}"
    print(tplt.format("排名","学校排名","分数",chr(12288)))
    for i in range(num):
        u=ulist[i]
        print(tplt.format(u[0],u[1],u[2],chr(12288)))


def main():
    unifo = []
    url = 'http://www.zuihaodaxue.cn/zuihaodaxuepaiming2020.html'
    html = getNetText(url)
    List(unifo,html)
    printList(unifo,40)
main()
```

2)情感分析代码

```
import numpy as np
import jieba


def open_dict(Dict = 'SCWX',
path=r'C:/Users/SCWX/Desktop/Textming/Textming/Sent_Dict/Hownet/'):
    path = path + '%s.txt' % Dict
    dictionary = open(path, 'r', encoding='utf-8')
    dict = []
    for word in dictionary:
        word = word.strip('\n')
        dict.append(word)
    return dict

def judgeodd(num):
    if (num % 2) == 0:
        return 'even'
```

```
    else:
        return 'odd'

deny_wordopen_dict(Dict='Deny',path=
r'C:/Users/SCWX/Desktop/Textming/Textming/')
posdict=open_dict(Dict='positive',path=
r'C:/Users/SCWX/Desktop/Textming/Textming/')
negdictopen_dict(Dict='negative',path=
r'C:/Users/SCWX/Desktop/Textming/Textming/')
degree_word=open_dict(Dict='phraseoflevel',path=
r'C:/Users/SCWX/Desktop/Textming/Textming/')
mostdict=degree_word[degree_word.index('extreme')+1                              :
degree_word.index('very')]#权重 5
verydict = degree_word[degree_word.index('very')+1 : degree_word.index('more')]#
权重 4
moredict = degree_word[degree_word.index('more')+1 : degree_word.index('ish')]#
权重 2
ishdict = degree_word[degree_word.index('ish')+1 : degree_word.index('last')]#
权重 0.5
def sentiment_score_list(dataset):
    seg_sentence = dataset.split('。')
    count1 = []
    count2 = []
    for sen in seg_sentence: #循环评论
        segtmp = jieba.lcut(sen, cut_all=False)  #句子分词
        p1 = 0 #记录位置
        p2 = 0 #
        negcount = 0
        negcount2 = 0
        negcount3 = 0
        poscount = 0 #积极词初始分值
        poscount2 = 0 #积极词反转后分值
        poscount3 = 0 #积极词最后分值
        for word in segtmp:
            if word in posdict:  #判断是否是情感词
                poscount += 1
                c = 0
                for w in segtmp[p2:p1]:  #扫描程度词
                    if w in mostdict:
                        poscount *= 5
                    elif w in verydict:
                        poscount *= 4
                    elif w in moredict:
                        poscount *= 2
```

```
            elif w in ishdict:
                poscount *= 0.5
            elif w in deny_word:
                c += 1
        if judgeodd(c) == 'odd':  #扫描否定词数
            poscount *= -1.0
            poscount2 += poscount
            poscount = 0
            poscount3 = poscount + poscount2 + poscount3
            poscount2 = 0
        else:
            poscount3 = poscount + poscount2 + poscount3
            poscount = 0
        p2 = p1 + 1

    elif word in negdict:  #分析同上
        negcount += 1
        d = 0
        for w in segtmp[p2:p1]:
            if w in mostdict:
                negcount *= 5
            elif w in verydict:
                negcount *= 4
            elif w in moredict:
                negcount *= 2
            elif w in ishdict:
                negcount *= 0.5
            elif w in degree_word:
                d += 1
        if judgeodd(d) == 'odd':
            negcount *= -1.0
            negcount2 += negcount
            negcount = 0
            negcount3 = negcount + negcount2 + negcount3
            negcount2 = 0
        else:
            negcount3 = negcount + negcount2 + negcount3
            negcount = 0
        p2 = p1 + 1
    elif word == '！' or word == '!':  ##判断感叹号
        for w2 in segtmp[::-1]:  # 扫描情感词
            if w2 in posdict or negdict:
                poscount3 += 2
                negcount3 += 2
```

```
                    break
            p1 += 1

            pos_count = 0
            neg_count = 0
            if poscount3 < 0 and negcount3 > 0:
                neg_count += negcount3 - poscount3
                pos_count = 0
            elif negcount3 < 0 and poscount3 > 0:
                pos_count = poscount3 - negcount3
                neg_count = 0
            elif poscount3 < 0 and negcount3 < 0:
                neg_count = -poscount3
                pos_count = -negcount3
            else:
                pos_count = poscount3
                neg_count = negcount3

            count1.append([pos_count, neg_count])
        count2.append(count1)
        count1 = []

    return count2

def sentiment_score(senti_score_list):
    score = []
    for review in senti_score_list:
        score_array = np.array(review)
        Pos = np.sum(score_array[:, 0])
        Neg = np.sum(score_array[:, 1])
        score.append([Pos, Neg])
    return score
print(sentiment_score(sentiment_score_list(‘代入相关数据 data 集’)))#data 数据请见支撑材料
```

3 .神经网络预测代码

```
function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Auto-generated by MATLAB, 24-May-2020 11:04:07.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
%   X = 1xTS cell, 1 inputs over TS timesteps
```

```
%   Each X{1,ts} = Qx3 matrix, input #1 at timestep ts.
%
% and returns:
%   Y = 1xTS cell of 1 outputs over TS timesteps.
%   Each Y{1,ts} = Qx1 matrix, output #1 at timestep ts.
%
% where Q is number of samples (or series) and TS is the number of timesteps.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = [0.05;0;0];
x1_step1.gain = [0.0004415059768871 62;1.60128102481986e-06;8.302200083022e-07];
x1_step1.ymin = -1;

% Layer 1
b1                                                                                  =
[3.2821562481286519208;2.5652125890994494917;-12.760003573436010882;-2.5293069607428151357;0.18016078216676192914;0.020110914249370080709;1.7831512095480788194;-0.98887199237696898901;2.3569080389610337356;3.0246049846121203508];
IW1_1          =          [-0.86478663514054021633          -1.3487414666560377796
-2.2149763798657113867;-1.2982819160171665818          -2.5810216691132747968
-0.074933321972915745146;0.83756753454313426221          2.9081440148974273541
-15.544640570153234194;2.6001299329293003915          0.42056715762606750042
-0.1041167134010875378;1.6484877185127391197          -0.72237153353289196289
2.6355710480338041535;-1.0588522276021097301          -3.0015683528960468429
-1.0286927392281628446;-0.84183993196422779448          3.0918711949109445314
2.7946890736878930994;1.2614282397045453177          4.3798713994759594925
-5.2803289816154448388;0.24561354542171068283          -2.9642475266093177844
-0.36302840285289028621;0.1772104657307965736          -2.1642346049113161399
-2.0756139687885615253];

% Layer 2
b2 = -0.764191093408990052;
LW2_1 = [-1.0204638216697392572 -1.3388029402316321548 -5.9433561340251790384
0.68069768434508926003       -1.6873962851151671494       0.87423853392028694209
1.4825343498032610423       2.2280286258414938594       -0.33181110314586814702
-0.28377962594407524222];

% Output 1
y1_step1.ymin = -1;
y1_step1.gain = 0.666666666666667;
```

```
y1_step1.xoffset = 1;

% ===== 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

% ===== MODULE FUNCTIONS ========

% 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
```