

网络社交传播视域下动态“信息茧房”情绪演化研究

摘要

针对目前网络社交传播格局中存在的“尖叫效应”、“回声室效应”与“信息茧房”传播困境等现实情况,本文深入了解了舆情传播的机理与影响因素,并针对传播现实探究了舆情传播机制中各类问题产生的底层逻辑,通过构建合理有效的舆情影响因素评价体系对产生上述问题的原因进行了深度剖析,在此基础上制定了破除“尖叫效应”、“回声室效应”、规避“信息茧房”的策略,并形成一篇针对网络社交“信息茧房”困境治理路径研究的专题报告。

针对问题一:通过构建基于 **LDA-BiLSTM** 的舆情演化趋势分析模型对“堕胎”和“控枪”两个话题进行了定量的机制分析,并分析了产生影响的关键因素。首先,本文运用 Python 从 8 款社交网络应用中爬取了 20000 条相关话题的评论文本,经过数据清洗最终保留了 13962 条有效文本,然后,通过采用 **LDA** 模型对语料文本进行困惑度计算、主题数量选择和 **LDA** 主体选择,最终确定选定话题总结凝练为 5 个主题标识。最后,本文使用 **BiLSTM** 方法对各个主题标识对应的评论文本进行情感分析,得到“堕胎”话题最终观点趋于相同(中立观点),而“控枪”话题最终观点趋于两极分化(观点极化)。

针对问题二:首先,综合考虑互联网舆情传播现实,本文在 **SIR** 传染病模型的基础上构建了 **SMPN-互联网情绪传播模型**,将网络用户分为中立立场(**M**)、积极立场(**P**)、消极立场(**N**)和易感状态(**S**)四类,通过仿真模拟互联网用户在舆情演化过程中的情绪变化,验证了观点极化与中立观点形成的六点假设。其次,通过构建网络演化模型对舆情传播网络进行了动态仿真,探索“尖叫效应”、“回声室效应”与“信息茧房”效应的形成机制。最后,本文通过对相关文献进行回顾并结合舆情传播实际,建立了包含 5 个指标的舆情演化影响因素评价模型。这 5 个指标分别为话题吸引力、用户活跃度、用户心理、不同用户相互影响力与平台推荐算法。通过层次分析法,结合 5 个指标,对舆情演化影响因素的重要性进行了评价。

针对问题三:基于问题二建立的舆情演化因素评价模型,本文从话题质量水平、用户传播水平与媒介影响水平三个维度出发改善舆情传播现状。在话题质量水平方面,应加强文化价值引领,积极提升网络话题质量;在用户传播水平方面,网络用户应提高媒介素养与信息甄别的能力;在媒介影响水平方面,主流媒体应加强智能技术应用的行业规范,坚持公益属性与价值理性。

针对问题四:本文在上述数据分析与数学模型的基础上,提出应积极推进政府、媒体与网民三方共同发力,以行业规制提高“信息茧房”治理效率,以行业自律完善“信息茧房”治理模式,以社会个体推进“信息茧房”治理深入,从而营造好清朗的网络空间,破局“信息茧房”传播困境。

关键词: LDA-BiLSTM 情感分析 SMPN-互联网情绪传播模型 网络演化模型 层次分析法

一、问题重述

1.1 问题背景

在现代媒体环境中，舆情传播具有潮汐性、快消性和不确定性等特征。随着大数据和人工智能的进一步发展，网络社交媒体逐渐成为舆情信息产生与传播的重要阵地，并产生了一系列亟待解决的现实问题。面对新形势下“尖叫效应”、“回声室效应”与“信息茧房”等信息传播困境的出现，如何破局并使网络舆论环境更具理性和建设性，成为了当前信息传播领域的重要任务。

1.2 需要解决的问题

问题一：针对特定话题的传播过程进行定量的机制分析，并分析舆情传播过程中的关键影响因素。在数据分析过程中，需要针对至少两种不同的话题，应包含中立共识与观点极化两种可能情况。

问题二：针对问题一中中立共识和观点极化两种可能情况，分析其产生的作用机制。对“尖叫效应”、“回声室效应”与“信息茧房”形成机制进行讨论，并分析话题西硬度、活跃度等因素对信息传播困境所造成的影响。

问题三：在问题二的基础上，提出破除“尖叫效应”和“回声室效应”、规避“信息茧房”困境的切实有效的策略方案。

问题四：结合上述问题的思考，从政府规划、媒体引领与网民自觉三个维度出发、针对如何规避“信息茧房”提出合理的解决方案与建议，并形成 1-2 页专题调研报告。

二、模型假设

- 假设社交媒体用户能够详细获取网络话题的相关信息；
- 假设新闻媒体对话题的报道均为公正客观的；
- 假设在舆情传播过程中未发生屏蔽话题等导致舆情传播强制中断的情况；
- 不考虑社交媒体之外的舆情传播渠道对用户认知的影响。
- 假设针对一个具体话题，所有的用户均处于一个封闭的系统中，总人数有限且不变。

三、符号说明

符号	说明	单位
$topic$	语料处理结果的主题标识	-
$S(t)$	t 时刻易感人群比例	
$M(t)$	t 时刻持有中立立场人群的比例	
$P(t)$	t 时刻持有积极立场人群的比例	
$N(t)$	t 时刻持有消极立场人群的比例	
$Topic_u$	用户 u 当天点赞、转发与评论的微博数量	
$Friend_u$	用户 u 在网络社交媒体中的好友人数	

四、模型的建立与求解

4.1 问题一模型的建立与求解

4.1.1 问题一的分析

问题一要求我们针对网络社交媒体中针对特定问题的评论语料文本，分析特定话题的网络舆情演化趋势，并通过处理演变过程中包含的定量特征，形成舆情演化趋势的分析模型，讨论其影响因素。

对此，我们建立了基于 LDA-BiLSTM 舆情演化趋势分析模型：首先，我们利用 Python 语言进行编码，针对网络社交媒体中关于堕胎和控枪等话题的评论预料文本进行爬取。然后，利用预处理后的语料数据，我们通过基于 LDA 的主题模型法和基于 BiLSTM 的情感分析法进行堕胎和控枪特定话题的主题表示和关键词抽取，并运用 BiLSTM 对分类后评论语料进行情感倾向分析和情感分布统计，得到每个主题下的正/负向分布情况。最后，通过舆情演化趋势结果分析可得，“堕胎”最终观点趋于相同（中立共识），而“控枪”最终观点趋于两极分化（观点极化）。

4.1.2 问题一模型的建立

(1) 基于 LDA 的主题模型

LDA 主题模型是无监督机器学习领域用于文档建模的统计模型。LDA (Latent Dirichlet Allocation, LDA) 这一主题模型被广泛用于文献主题发现、用户推荐、话题演化等领域。本文基于 LDA 主题模型对堕胎和控枪等话题的评论预料文本进行主题抽取，在 LDA 模型中，主题产生过程可分解为以下 3 个过程。

1) **词向量化**。对数据预处理后文本进行分析，将分词后的文本输入到 Word2vec 模型中，利用 Skip-gram 方法进行词向量化训练，得到文本的词向量化文件。

2) **主题数量选择**。LDA 主题模型的评价标准困惑度 (Perplexity) 已在 LDA 模型中广泛用于确定主题的数量。本文计算了由问题组成的 D 个文档的困惑度，以确定文档潜在的主题数量。

3) **LDA 主题抽取**。LDA 主题生成模型将文档的主题混合权重视为 D 维随机变量，并将每个主题的单词权重作为具有狄利克雷先验分布的 k 维随机变量。其中 D 是文档数， k 是主题数量。

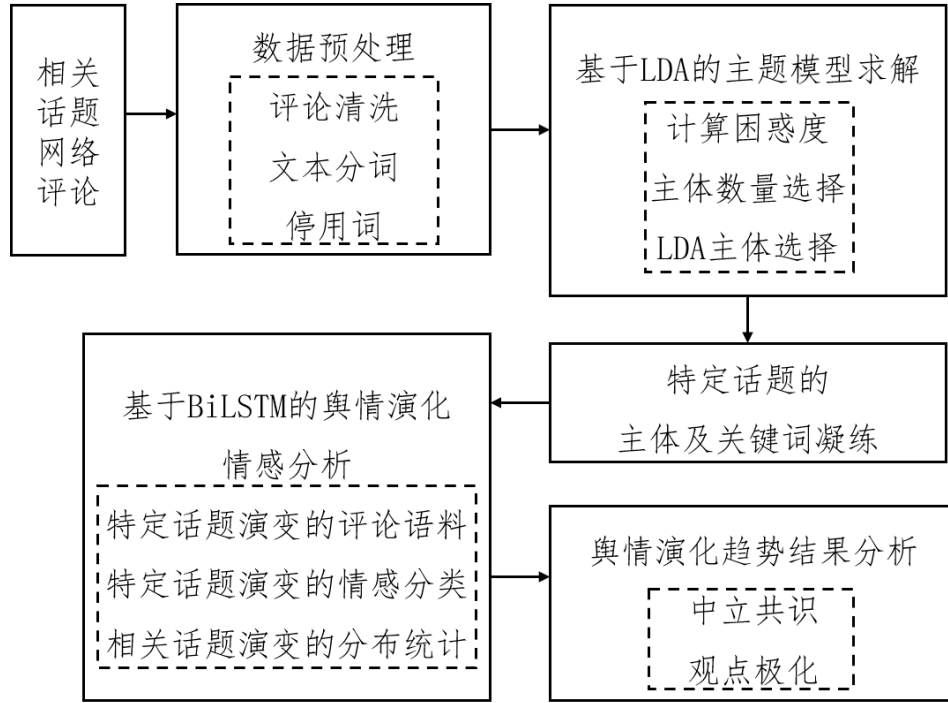


图 1: 舆情演化分析模型框架

文档的主题向量用 θ 表示，见公式 (1)。将权重最大的主题作为该文档的主题，并以此作为最终的服务质量分类依据。

$$\theta = (topic_1, weight_1), topic_2, weight_2), \dots, (topic_z, weight_z) \quad (1)$$

(2) 基于 BiLSTM 的情感分析

双向长短时记忆模型 (Binary Long Short Term Memory, BiLSTM) 是一种双向递归神经网络，它将整个句子的所有单词作为输入，充分考虑了文本的上下文信息，使得信息从前往后和从后往前两个方向进行传递。BiLSTM 模型既能弥补 LDA 模型在短文本数据稀疏方面的不足，又能记忆上下文信息，更好地学习文本特征，可以很好地处理多条短文本语料。本文运用 Bi-LSTM 对 LDA 模型得到的各主题对应的评论文档进行情感倾向分析，并对对应主题的评论语料的情感分布进行统计，得到每个分类主题下的正/负向的文档分布情况。

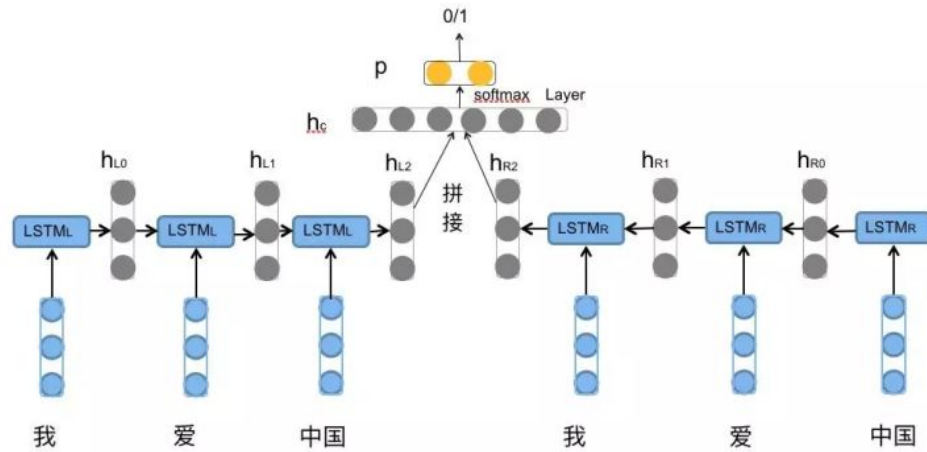


图 2: BiLSTM 运行机制图

4.1.3 问题一模型的求解

由于不同的社交网络媒体具有一定差异化的用户属性,所以我们针对网络社交媒体中关于“堕胎”和“控枪”等话题,从微博、知乎和小红书等 8 款社交网络应用中爬取总计 20000 条相关话题评论文本(“堕胎”相关评论文本 10000 条,“控枪”相关评论文本 10000 条,时间跨度从 2022 年 7 月 25 日至 8 月 20 日),以期最大程度避免由于特定网络应用中用户的情感倾向对实验结果造成的误差。在剔除无效评论(如英文评论、繁体字评论、无意义评论等)和删除重复评论数据后,最终收集了 13962 条评论数据(“堕胎”相关评论文本 6724 条,“控枪”相关评论文本 7238 条)进行主题挖掘和细粒度情感分析。网友在线评论文本内容如下列图表所示。

路径	网友评论(以#堕胎#话题为例)
https://www.zhihu.com/question/539543493/answer/2544016847	明确了上述事实,就知道美国的堕胎权争议根本就是美国特色的问题,对西欧的影响都有限,更不用说异质文明国家了。那些把美国当作全世界,美国人权倒退就要拉着全世界一起反思的殖人们可以省省了。另
https://www.zhihu.com/question/539543493/answer/2544783156	决定堕胎是否合法的权利,将下放到各州自行决定。
https://www.zhihu.com/question/539543493/answer/2544543153	强行限制自由选择的权利,反而是对生命的不尊重。以反对堕胎为名,压制人类自由的天性本身就是一个谬论怪圈。
https://www.zhihu.com/question/539543493/answer/2543862975	最高法院的这项最新裁决将赋予各州制定堕胎法的权力,预计几乎一半的州将宣布堕胎为非法或严格限制堕胎。法院意见书称:“宪法没有赋予堕胎权;‘罗诉韦德案’裁决被驳回;规范堕胎的权力还给了人民和他们选出的代表。”
https://www.zhihu.com/question/539543493/answer/2545256535	如果不是因为男人,女人为什么要去堕胎? 所以我建议美国人立法,每次有女性堕胎,就让搞大她肚子的那个男人去坐牢。
https://www.zhihu.com/question/539543493/answer/2543992824	支持反对的各州划分成两个派别,中部普遍反对堕胎,东西部发达地区普遍支持,堕胎法案通过法律形式把这两派的对立长期化,固定化。要解决问题未来要再次修法,这又是难于上青天的事不知道未来会不会发生,能不能实现,法律形式的长期效力将长久影响两派各州的对立,并且难以轻易缓和。
https://www.zhihu.com/question/539543493/answer/2544551608	他们选择放宽堕胎的那一天就应该知道这一天迟早会来。一个人口不断萎缩的群体怎么可能永远占据民意的大多数?

图 3: 网络评论文本示例

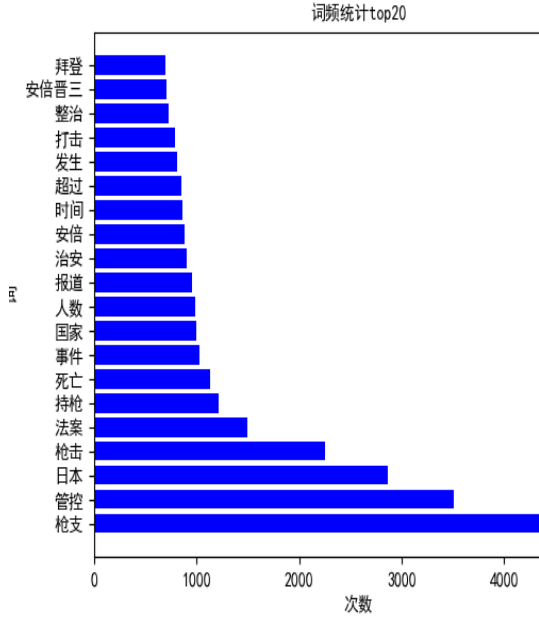


图 4: “控枪”话题词频图

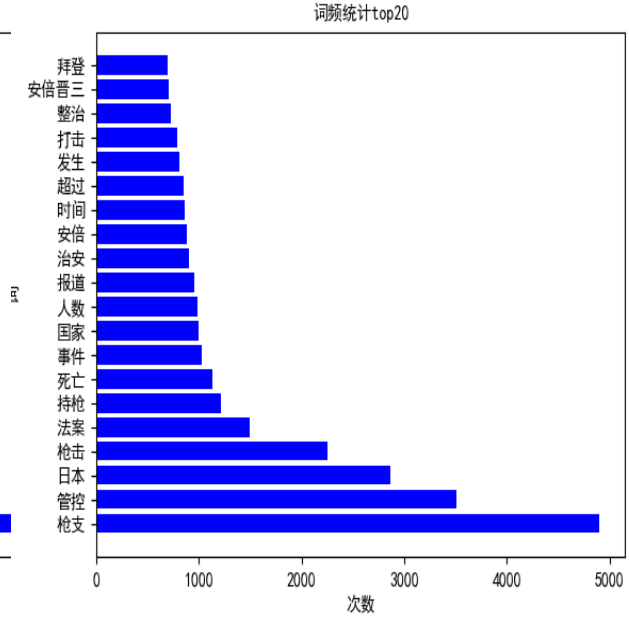


图 5: “堕胎”话题词频图

首先，以“控枪”话题为例，进行基于 LDA 主题模型求解的实验。

本文主要采用吉布斯采样方法实现 LDA 模型超参数估计。其中超参数 α 和 β 分别设置为对称的狄利克雷先验参数，其值为 $50/T$ 和 0.01 。Gibbs 采样的迭代次数设置为 100，文档贡献度阈值设置为 $1/k$ 。LDA 模型的重要参数是潜在主题的数量。实验中的 LDA 模型的参数在每个主题数量 k 是相同的。同时本文采用困惑度确定 LDA 最优主题数目。

困惑度的的计算方式为:

$$perplexity = \exp - \frac{\sum_{d=1}^D \log P(W_d)}{\sum_{d=1}^D N_d} \quad (2)$$

其中 w_d 表示出现在文档 d 中的单词， N_d 表示文档 d 中单词的数量， $P(W_d)$ 是指每个单词出现在语料库 D 中的概率。

在此次实验中，测试了 30 个主题数值，并且在主题数量方面进行了困惑度曲线拟合，同时结合 LDAvis 可视化结果发现，当主题数 $K=5$ 时，主题之间的重叠较少，分类效果较好最终确定了 5 个主题为最佳解决方案，所以每条评论由 5 个主题向量组成。

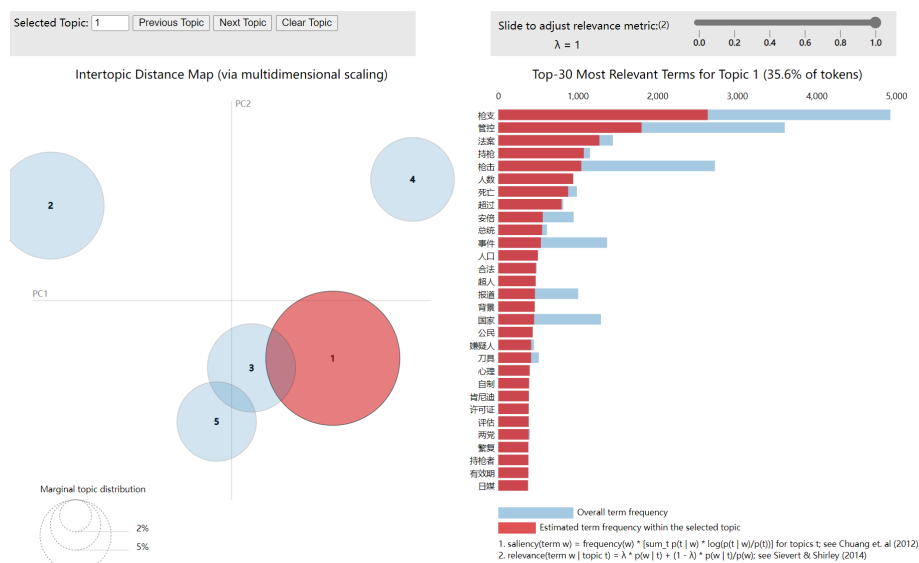


图 6: pyLDA 可视化

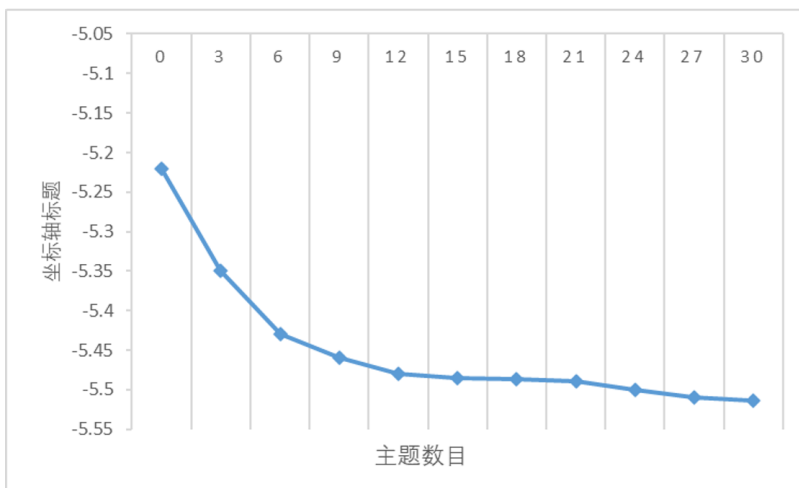


图 7: 困惑度计算结果

通过 LDA 主题分类得到“控枪”话题网络评论语料的 5 个主题以及每个主题的词项分布，将每个主题下排名前 5 的高概率特征词进行整理，并依据每个主题下的高概率特征词进行场景描述，总结归纳出最符合当前主题下高概率特征词的主题标识。例如在 Topic1 中，枪支、管控和法案等高概率特征词与控枪为目的立法规制较为密切，因此将 Topic1 定义为立法规制枪支使用。同理，最终将“控枪”话题网络评论语料归纳结果分为 5 类，如下表所示。

编号	主题标识类别	主题的前5个高频率特征词
Topic1	立法规划枪支使用	枪支、管控、法案、持枪、枪击
Topic2	控枪保障公民安全	整治、枪支、安全、伤亡、惊慌
Topic3	重点打击枪支犯罪	治安、打击、枪击、维稳、警方
Topic4	民众呼吁枪支管控	枪支、管控、呼吁、民众、安保
Topic5	纽约州限制枪支使用	枪支、管控、纽约州、大规模、限制

图 8: 主题-高概率特征词分布

在运用 LDA 模型进行主题表示划分后, 本文使用基于 BiLSTM 方法对各个主题标识对应的评论文本进行情感倾向分析, 并在此基础上进行时间序列纵向趋势分析, 由可视化结果(如下图所示)可得, "堕胎" 最终观点趋于相同(中立共识), 而"控枪" 最终观点趋于两极分化(观点极化)。

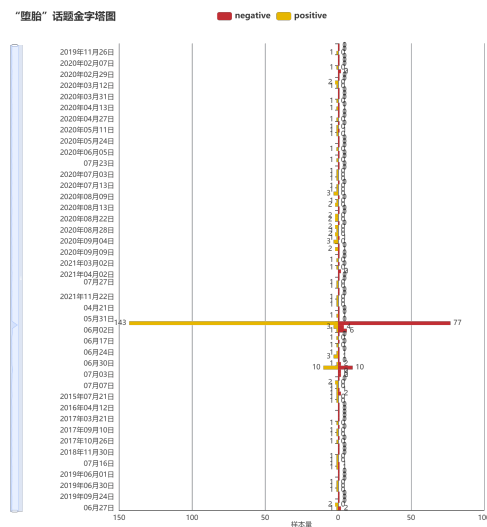


图 9: “堕胎” 话题舆情演化趋势

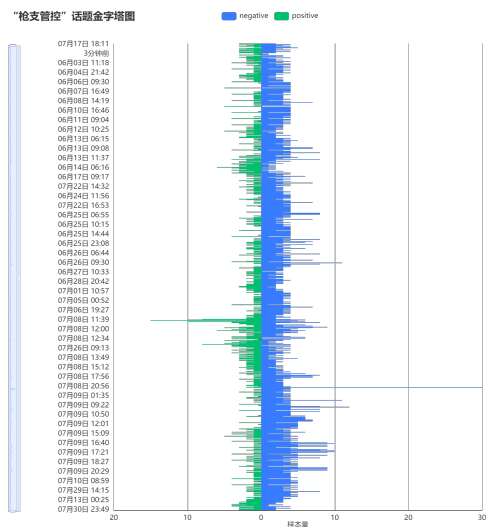


图 10: “控枪” 话题舆情演化趋势

4.2 问题二模型的建立与求解

4.2.1 问题二模型的分析

中立观点与观点极化现象的产生, 受到了用户评价所处的情绪状态, 用户态度间情绪的相互影响与用户观点转化规则等因素的影响。

在 SIR 模型中, 人群被分为了易感人群 (S)、患病人群 (I)、移除人群 (R) 三类。易感人群是未患病, 但与患病者接触后容易感染的人群, 移除人群是患病后痊愈并不会再被感染的人群。我们认为, 互联网中用户对某一具体事件的态度被分为三类, 分别是中立立场 (M)、积极立场 (P)、消极立场 (N), 而还未接触这一事件的人, 由于对事件并不了解, 没有具体态度, 还处于易感状态 (S)。基于此, 我们构建了 $SMPN$ -互联网情绪传播模型。

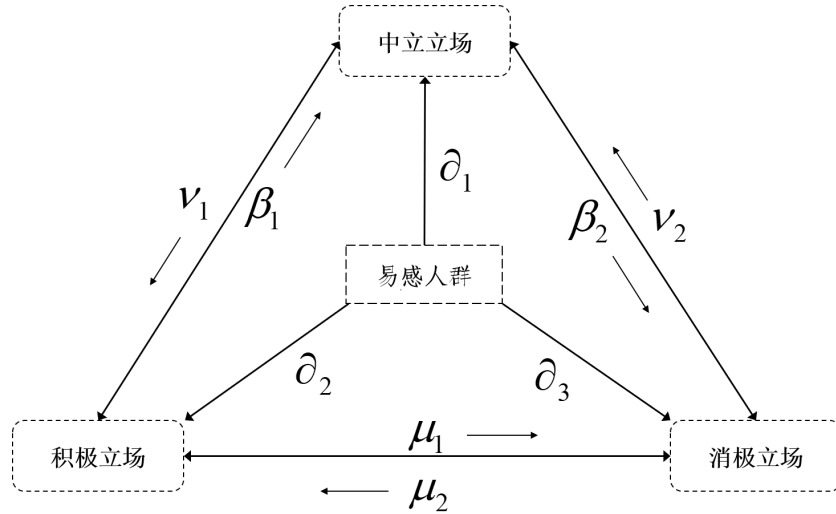


图 11: *SMPN*-互联网情绪传播模型

如上图所示，在 *SMPN*-互联网情绪传播模型中，易感人群和持有中立立场、积极立场、消极立场的人群比例在 t 时刻分别为 $S(t)$ 、 $M(t)$ 、 $P(t)$ 、 $N(t)$ ，易感人群有 α_1 的概率成为中立立场人群，有 α_2 的概率成为积极立场人群，有 α_3 的概率成为消极立场人群。 β_1 、 β_2 、 γ_1 、 μ_1 、 γ_2 、 μ_2 分别表示持有不同立场的网络用户相互转化的概率。

在通过 *SMPN*-互联网情绪传播模型对中立观点和极化观点产生机制的定量模型构建的基础上，通过构建网络演化模型并运用 AHP 综合评价方法对产生“尖叫效应”、“回声室效应”与“信息茧房”舆情困境的影响因素做出了进一步的研究。

4.2.2 问题二模型的建立

由于人的情绪和立场是多变的，一个人对某一具体时间的态度随时可能受到互联网中其他用户情绪状态的影响，因此我们认为互联网中用户的情绪没有“痊愈”状态，这是区别于传染病传播模型的第一点。其次，在 *SIR* 中，只有一种传染病，也即患病人群只有一种，而在 *SMPN* 中，互联网用户被区分为了三类，他们持有不同的立场，并可以通过与其他立场的互联网用户的接触，通过一定的概率发生情绪状态的改变。

(1) *SMPN*-互联网情绪传播模型

在话题产生初期，易感人群接触话题后会产生自己主观的第一立场，他们或对此一事件持积极态度，或持消极态度，或持中立态度。随着话题的传播，互联网用户会不断地传递自己的情绪，他们也会在主动或被动地接受他人立场的影响中随时发生立场的转变，即传染方向不固定。在一段时间后，互联网封闭系统中所有用户针对这一具体话题的立场最终会达到相对稳定状态。首先，运用 *SMPN*-互联网情绪传播模型对中立观点和观点极化的产生机制提出以下六点假设：

观点极化的产生是由于：

假设一：中立立场者容易受到极端立场者的影响并转变自己立场；

假设二：极端立场者不容易受到中立立场者，不容易转变立场；

假设三：易感人群转换成极端立场人数比例大；

中立观点的产生是由于：

假设四：极端立场者容易受到中立立场者的影响并转变自己立场；

假设五：中立立场者不容易受到极端立场者的影响，不容易转变立场；

假设六：易感人群转换成中立立场人数比例大；

通过综合分析舆情传播中观点极化与中立观点产生的特征，建立了以下情绪传播的微分方程，其中左式表示 t 到 $t+\Delta t$ 时刻内，各状态下的互联网用户人数变化。

$$\begin{cases} \frac{dS(t)}{dt} = -\alpha_1 SM - \alpha_2 SP - \alpha_3 SN \\ \frac{dM(t)}{dt} = \alpha_1 SM - \beta_1 MP - \beta_2 MN + \gamma_1 MP + \gamma_2 MN \\ \frac{dP(t)}{dt} = \alpha_2 SP - \beta_1 MP - \gamma_1 MP - \mu_1 PN + \mu_2 PN \\ \frac{dN(t)}{dt} = \alpha_3 SN - \beta_2 MN - \gamma_2 MN + \mu_1 PN - \mu_2 PN \end{cases} \quad (3)$$

其中初始参数为： $S(0)=0.6$, $M(0)=0.05$, $P(0)=0.2$, $N(0)=0.15$, 时间 $t \in [0, 50]$ 。实验组与对照组保持对照和单一变量（同组参数）原则，由于参数之间存在等式关系，参数成对变化。在出现两组参数同时改变时，保持实验组和对照组的相同两组参数比例相同（保留三位小数）。

通过模拟网络的生成过程，能够深入了解舆情传播过程中信息传播的演化趋势，从而针对“尖叫效应”、回声室效应”与“信息茧房”的产生机制进行研究。综合考虑数据来源特征，选择 BA 无标度网络模型进行研究

(2) 网络演化模型

模拟网络的生成过程可以更好的了解网络演化的机制。常见的网络演化模型有随机网络模型，小世界网络模型，无标度网络模型。考虑到本文建模运用的文本数据多来源于微博、知乎等社交网络媒体，从而选择有向加权、幂律分布的 BA 网络作为舆情传播演化的模型基体。

基于此，本文构建了舆情传播网络演化模型（如下图所示），并对“尖叫效应”，“回声室效应”与“信息茧房”困境的形成机制进行了综合分析。

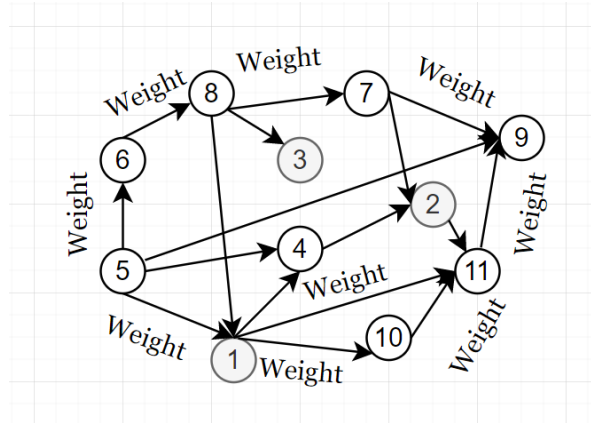


图 12: 网络演化模型示意图

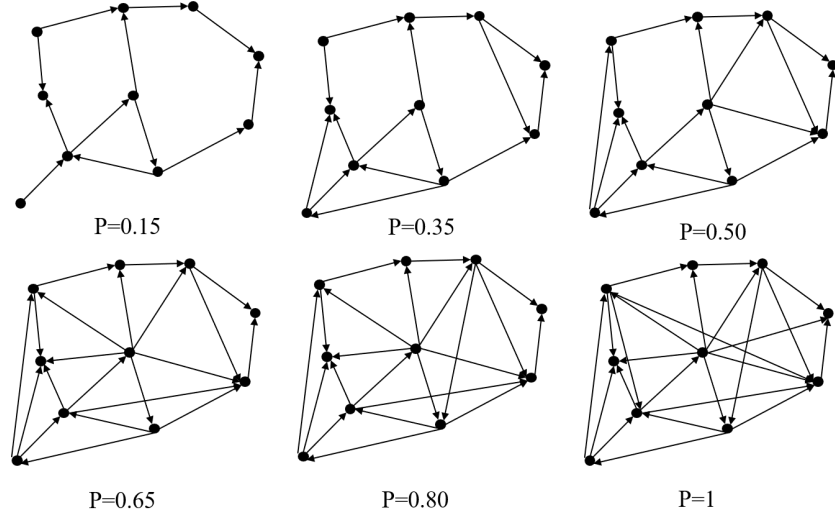


图 13: WS 小世界模型

(3) 基于层次分析法的舆情演化影响因素评价模型

评价指标的选取与计算:

参考国内外已有研究,发现在现有研究中多从“网络用户”、“舆情传播者”与“政府监管”等角度出发,探讨信息传播环境的重要影响。通过深入挖掘提取网络信息传播中相关话题的评价信息,我们分别从话题质量水平 (C_1)、用户传播水平 (C_2) 和媒介影响水平 (C_3) 这三个方面进行细化指标选取,从而选取话题吸引度 (A_1)、用户活跃度 (A_2)、用户心理 (A_3)、不同用户相互影响力 (A_4) 和平台推荐算法 (A_5) 5 个二级的细化指标用以评价舆情演化影响因素的影响能力,具体的细化指标计算公式如下:

1. 话题吸引度 (A_1)

为反映目标话题对于网络用户的吸引能力与热度,本文通过爬取目标话题下的网络用户转发、点赞评论的微博数量与其当日发布的带目标话题的微博总数,构造了话题吸引度指数,计算公式如下:

$$attraction(u) = \frac{T - Topic(u)}{Topic(u)} \quad (4)$$

2. 用户活跃度 (A_2)

为反映用户针对特定话题的活跃程度与传播积极性,本文通过爬取用户在特定时间段内转发、点赞和评论的微博总数,并进行加总求和,构造用户活跃度指数,计算公式如下:

$$activity(u) = transmit(u) + comment(u) + like(u) \quad (5)$$

3. 用户心理 (A_3)

为了解网络用户对新鲜话题的普遍接受程度与开放程度,本文通过计算用户浏览记录的历史相似性进行定量化描述,计算公式如下:

$$ability(u) = n^{-1}(n-1)^{-1} \sum_i^n \sum_j^n R(t_i, t_j) \quad (6)$$

其中 i, j 表示任意两条符合条件的微博, $R(t_i, t_j)$ 表示 t_i, t_j 两条微博之间的余弦值。

4. 不同用户相互影响力 (A_4)

根据 $SMPN$ -互联网情绪传播模型可知, 不同网络用户之间存在着相互影响, 相互作用的关系, 设同时存在用户 u 和用户 v , $friend(u)$ 表示为用户 u 在网络社交媒体中的好友人数, 且 n_f 为用户 u 的好友人数, 则用户 u 相对于用户 v 的影响力计算公式如下:

$$affection(u) = n_f^{-1} \sum_{v_i}^n \frac{|friend(u) \cap friend(v)|}{|friend(u)| + |friend(v)|} \quad (7)$$

5. 平台推荐算法 (A_5)

网络社交平台以用户浏览偏好与阅读习惯为依据, 在新闻舆情推荐中充分考量历史记录相似性等特征, 本文通过用户微博历史浏览话题与平台推荐话题的相似度来衡量平台推荐算法的偏好程度, 计算公式如下:

$$algorithm(u) = n_1^{-1} n_2^{-1} \sum_i^n \sum_j^n R(t_i, t_j) \quad (8)$$

其中 n_1 为用户 u 微博的历史浏览数量, n_2 为网络平台推荐话题数量。

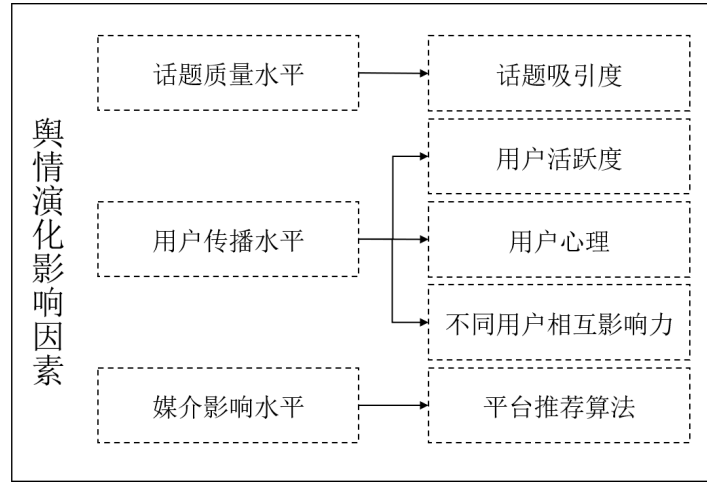


图 14: 评价指标体系

由上述的舆情传播影响因素评价指标体系可以看出, 舆情传播影响因素的评估是多因一果的过程, 对结果产生影响的指标较多, 并且各指标设立中存在着一定的联系。因此, 本文构建了基于层次分析法的评价模型, 以此对舆情传播影响因素进行评估。

层次分析法 (AHP) 将一个复杂的目标决策问题 (比如去哪旅游) 作为一个系统, 将目标分解为多个目标或准则, 通过定性指标进行相关计算, 以作为多方案优化决策依据的系统方法。简单来说就是将决策总是有关的元素分解成目标、准则、方案等层次, 在此基础上进行定性和定量分析的决策方法。

- (1) 对原始数据进行标准化处理;
- (2) 构建判断矩阵;

在成对比较的基础上，本文确定了各影响因素的优先顺序。根据各主要因素的重要性可确定其满足以下关系：

$$A_1 > A_2 \geq A_3 \geq A_4 > A_5 \quad (9)$$

根据上述优先顺序，可得判断矩阵 $(b_{ij})_{6 \times 6}$ ：

$$\begin{bmatrix} 1 & \frac{1}{3} & \frac{1}{5} & \frac{1}{7} & \frac{1}{9} \\ 3 & 1 & \frac{1}{3} & \frac{1}{5} & \frac{1}{7} \\ 5 & 3 & 1 & \frac{1}{3} & \frac{1}{5} \\ 7 & 5 & 3 & 1 & \frac{1}{3} \\ 9 & 7 & 5 & 3 & 1 \end{bmatrix} \quad (10)$$

(3) 层次单排序和一致性检验

运用和积法计算判断矩阵中针对其准则的相对权重，并进行一致性检验，公式如下所示：

$$CI = \frac{\lambda_{max} - n}{n - 1} \quad (11)$$

$$CR = \frac{CI}{RI} \quad (12)$$

(4) 汇总结果得到权重矩阵

4.2.3 问题二模型的求解

(1) SMPN-互联网情绪传播模型

基于以上的模型建立，我们设置初始参数并在 matlab 中对 SMPN 模型进行模拟仿真。并通过设置对照验证了六大假设，证明了中立观点和观点极化的产生机制。

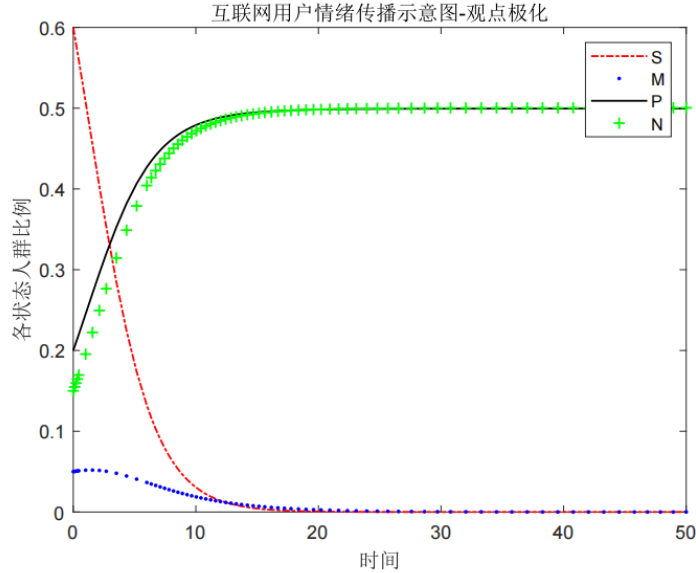


图 15: 观点极化

由图一，话题产生后，易感人群迅速接收话题，并形成自己的立场。随着话题传播，拥有极端立场的用户比例迅速增加，而拥有中立立场的用户比例逐渐下降。话题传播后期，各曲线趋于平稳，积极立场和消极立场的用户充斥了互联网，形成观点极化的状态。

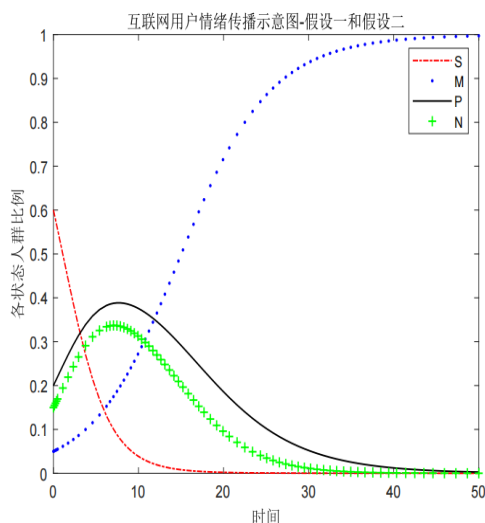


图 16: 假设一和假设二

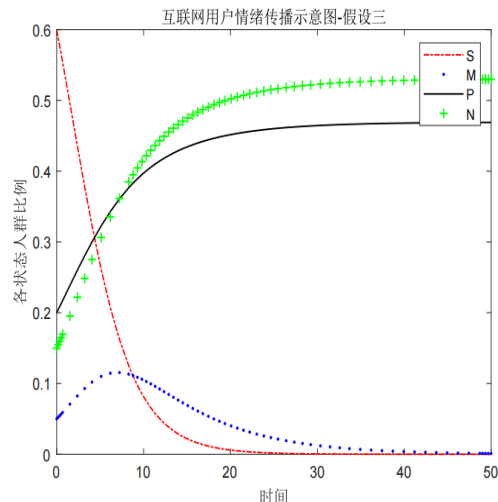


图 17: 假设三

假设一和假设二认为观点极化是由于极端立场和中立立场用户之间的转化率导致，观点极化是由于中立立场用户向极端立场用户转化的经转化率为正。通过实验组一我们可以发现，当逆转极端立场和中立立场用户之间的转化率，虽然极端立场用户比例短时间内呈上升状态，但最终在波峰后持续下降，而中立立场用户比例曲线呈“S”型持续上升。最终在 $t=40$ 后互联网处于相对稳定状态，而此时并非观点极化状态，验证了假设一和假设二的正确性。

假设三认为观点极化是由于易感人群转换成极端立场人数的比例大导致，即当话题本身具有争议性时，更容易形成观点极化状态。对比图一，图三在 $t=40$ 后才达到稳定状态，显然当减小易感人群转化为极端立场用户的比例时，会减缓极化状态的进程。如果话题没有持续发酵，那么可能最终互联网并没有处于观点极化状态，由此验证了假设三的正确性。

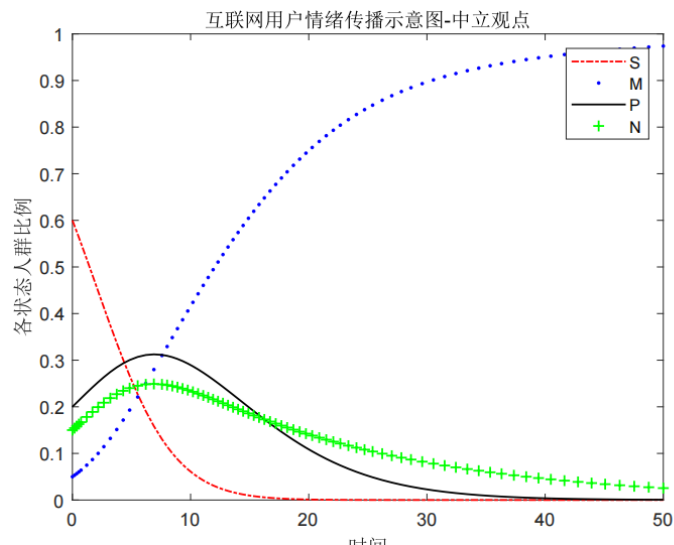


图 18: 中立观点

由图四，话题产生后，易感人群迅速形成自己的立场。话题传播阶段，拥有中立立场的用户比例持续增加，拥有极端立场的用户比例在经历一个小高峰后逐渐趋于 0。话题传播后期，各曲线趋于平稳，互联网中全部是中立立场的用户，形成中立观点的状态。

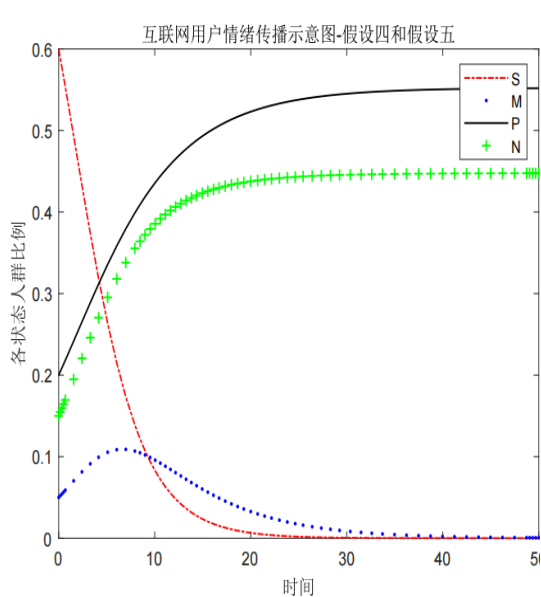


图 19: 假设四和假设五

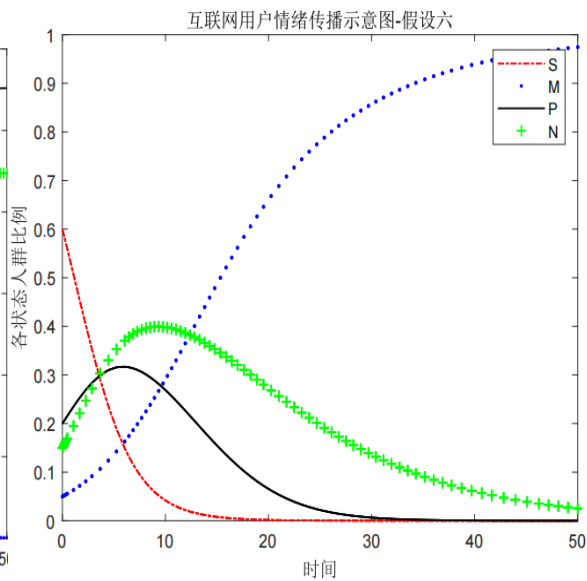


图 20: 假设六

假设四和假设五认为中立观点同样是由极端立场和中立立场用户之间的转化率导致，逆转极端立场和中立立场用户之间的转化率，我们会在实验组三中得到和实验组二相似的结论，即互联网最终状态也发生逆转，最终没有达到中立观点，因此假设四和假设五验证成立。

假设六认为中立观点是由于易感人群转换成中立立场人数的比例大导致，也即当话题本身不

具备争议性时，互联网更容易形成中立观点状态。对比图四，图六达到稳定状态速度更慢，原理同实验组二，**假设六的正确性也得证。**

(2) 网络演化模型

首先，根据网络文本来源的社交媒体特征，本文确定了 0 时刻网络兴趣空间的维度，例如微博社交网络中有 47 个互动版块，因而微博文本的兴趣空间维度定位 47；然后，针对选择的样本用户群体的社交媒体联系构建初始关系网，构建了一个包含 P_N 节点的强联通且非周期性的有向图，作为 0 时刻的微博网络，其中每个节点代表一个用户，有向图中的有向性代表了用户之间的联系。其次，从所有节点 P_N 中随机选取 P_M 个节点，对 P_M 中的每个节点，在一定范围内根据概率排序选取 PQ 个节点，并确定两两用户之间的相似度，概率计算原则为：

$$p = \frac{\text{degree}_Q}{\sum_j \text{degree}_j} \quad (13)$$

最后，通过分析舆情网络演化模型的集群性与各节点的平均距离，对“尖叫效应”、“回声室效应”和“信息茧房”困境的传播机理进行评价。其中，网络演化模型的集群性通过集群系数来反映，能够较好反映网络结构中用户联系的紧密程度，在无权无向网络中集群系数 C_i 的定义为：

$$C_i = \frac{2l_i}{k_i(k_i - 1)} \quad (14)$$

l_i 表示为与节点 i 的邻点存在的边的数目， k_i 为节点 i 的度。

而整个网络的集群系数等于所有节点集群系数的均值，即：

$$C = \frac{1}{n} \sum_{i=1} C_i \quad (15)$$

其中 n 为节点数。

网络中平均距离指的是所有节点最短距离的平均，无权网络中平均距离的计算公式如下所示：

$$l = \frac{1}{n(n-1)} \sum_{i \neq j} d_{i,j}(i, j = 1, 2, \dots, n) \quad (16)$$

$d_{i,j}$ 表示为节点 i 和节点 j 之间的最短距离。

在舆情传播网络演化模型的基础上，本文对“尖叫效应”，“回声室效应”与“信息茧房”困境的形成机制得出以下结论：

“尖叫效应”：

“尖叫效应”是指在舆情传播中，部分通过抓取、剪辑改编的低俗内容往往能更迅速地获取社交媒体用户的关注，通过满足用户的猎奇心理或是引起人们的强烈关注评论来获取高额的流量与点击量。在网络演化模型中集群系数呈现递减趋势，用户间的平均最短距离逐渐缩小，“尖叫”内容产生的效果不断扩大。

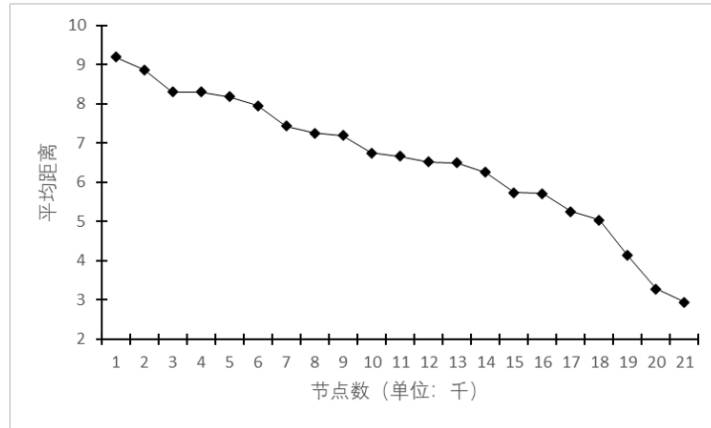


图 21: “尖叫效应”

“回声室效应”:

“回声室效应”是指在一个相对密闭的舆论环境中，只能重复听见一种声音，由于不断重复导致“回声”可能会被扭曲或放大。在网络演化模型中集群系数呈现递增趋势，用户间的平均最短距离逐渐增大，反映了在舆情演化过程中“回声”所代表的观点和信息呈现不断强化和扩散传播的趋势，并迫使其他舆情观点逐渐沉默。

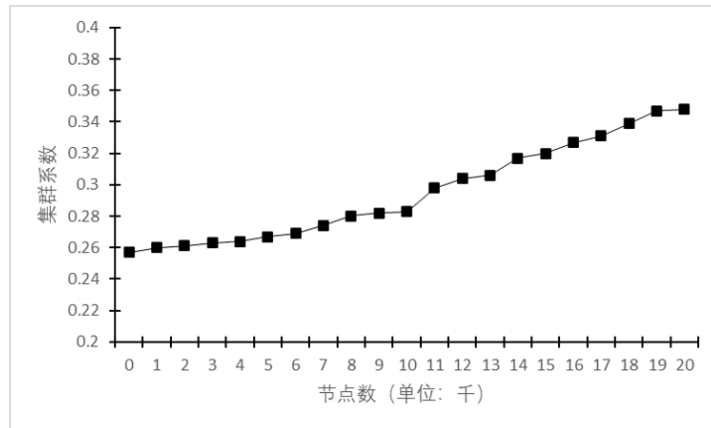


图 22: “回声室效应”

“信息茧房”:

在舆情传播过程中，“尖叫效应”与“回声室效应”使得网络社交媒体用户逐渐会自动过滤掉“不感兴趣”或“不认同”的信息，实现“看我想看，听我想听”的信息困境。久而久之，信息接收维度变窄，资讯获取渠道单一，在单调的信息交互中形成特定的兴趣习惯，从而深陷“信息茧房”之中。

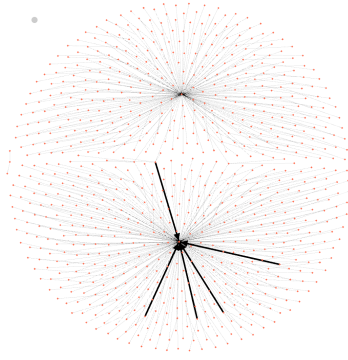


图 23: 网络演化模型集群性特征

(2) 舆情演化影响因素评价模型

根据层次分析法基本过程，运用 MATLAB 软件编程，对舆情演化影响因素评价指标体系进行分析，当 $n = 5$ 时， $RI = 1.24$ 。对于上述判断矩阵，可得 $CR = 0.0461 \leq 0.1$ ，因此该比较矩阵是合理的。

通过最大特征值所对应的特征向量，可得舆情演化影响因素的权重如下图所示：

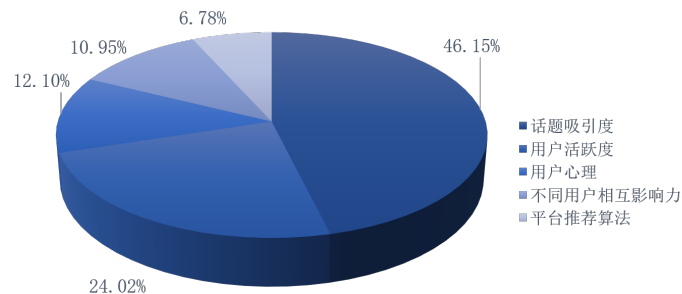


图 24: 舆情影响因素各指标权重

通过权重分析可知，话题质量水平、用户传播水平与媒介影响水平均对舆情演化产生了重要影响，其中话题质量水平对舆情演化影响最大，用户传播水平次之，媒介影响水平相较而言最小。

4.3 问题三的分析与求解

4.3.1 问题三的分析

在问题二中，本文对舆情演化过程中的重要影响因素进行分析，得出话题质量水平、用户传播水平与媒介影响水平对舆情传播效果产生了重大影响。其中 5 个二级指标中话题吸引力、用户活跃度、用户心理、不同用户相互影响力与平台推荐算法起到了重要的作用，对此，本文将以此为出发点探讨破除“尖叫效应”和“回声室效应”、规避“信息茧房”的策略。

4.3.2 问题三的求解

本文从话题质量水平、用户传播水平与媒介影响水平三个维度出发提出破除“尖叫效应”和“回声室效应”、规避“信息茧房”的策略：

(1) 话题质量水平

加强文化价值引领，积极提升网络话题质量。从网络文化的供给端、传播端与消费端三方共同发力，营造清朗的网络空间。宏观层面，应积极明确网络社区中舆情传播的标尺，对话题质量是否符合新时代舆情传播规范，从而为网民在舆情演化过程中的行为提供指导性建议；微观层面，网民与媒体作为舆论生态的“源头活水”，是提升网络话题质量，优化舆论传播范式的主力军。网络用户与主流媒体应在舆情传播规范的引领下努力推动网络话题质量的优化与改善。

(2) 用户传播水平

网络用户应提高媒介素养与信息甄别的能力。媒介素养是互联网用户对各种媒介传递的信息进行接收、判断、解读、理解和二次传播的能力，还包括互联网用户运用媒介接收和传播信息的能力。在纷繁复杂的网络社交环境中，增强明辨是非的能力有利于网络用户选择正向的、有价值的网络信息，为打破“信息茧房”构筑的单一信息源供给发挥重要作用。

(3) 媒介影响水平

主流媒体应加强舆论传播领域智能技术应用的行业规制与行业引导，坚持公益属性与价值理性推动行业可持续发展。平台推荐算法合理与否，关乎社交网络内容的合理性与可持续性。破除目前部分媒体大肆利用“尖叫效应”推送博人眼球的劣质低俗内容以获取关注和流量的困局，有利于抵制舆论生态的“劣币驱逐良币”，实现网络环境的风清气正。

五、网络社交“信息茧房”困境的治理路径研究

面向网络社会中信息过剩与智能算法对技术的影响,“信息茧房”在技术、传播、社会与个体的多重逻辑中出现了新的内涵变化与表现特征,并不断影响与规制虚实社会交往行为。面对“信息茧房”的负面效应,需要政府、媒体和网民三方共同发力,发挥政府行政规制效力,引导行业创新治理模式,激发社会个体参与治理的主动性

(一) 政府顶层设计: 行政规制提高“信息茧房”治理效率

从政府维度,需要加强对智能技术应用的正确引导,对算法推荐机制的不断规训与约束,加强政策导向的主动权,并强化对违法违规算法行为的打击力度。2022年3月《互联网信息服务算法推荐管理规定》开始施行,对算法推荐服务提供者提出具体规范,为用户的网络使用提出了权益保护措施,并从监督管理与法律效力的维度提出了国家层面的治理路径。这不仅有利于“信息茧房”的综合治理,也有利于对算法推荐服务所造成的多种不良行为进行规范引导。2022年4月,中央网信办“清朗·2022年算法综合治理”专项行动启动,这对于算法推荐引起的网络直播秩序混乱、大数据杀熟、用户陷入“信息茧房”、网络安全等进行综合系统治理。从部门规章到专项行动,是网络空间治理由宏观规划到具体落地的体现,通过行政权力规制互联网平台的网络信息服务、维护网民科学合理的信息接收空间,有利于增强网络空间协同治理的整体性和能动性。当然,在具体措施的执行与落地中,需要兼顾政策的针对性、在地化,技术的过度滥用与价值理性的权衡同时建构社会多元的公共网络平台引导机制、网络舆论监督与约束机制,营造良好的网络生态。

(二) 媒体价值引领: 行业自律完善“信息茧房”治理模式

从媒体或者平台维度,特别是主流媒体及头部的互联网平台,在利用媒介技术的基础上始终以公益属性放在首位,坚持公益性,以价值理性

驱动行业发展。在技术驱动传媒业的变革中,技术的作用至关重要,但是我们也应该以技术“善用”的思维去理解并利用技术。特别是以人工智能、算法推荐等媒体智能化发展趋势深入的当下,技术效能的发挥应该转向“内容变革—结构布局—治理模式”的发展路径,突出“以技术支撑治理”的思维逻辑,利用智能技术推动传媒行业、网络空间的治理现代化。从治理的维度看,“信息茧房”基于的算法推荐技术需要以治理现代化的视角进行完善。近期各主流媒体不断开展“主流算法”的做法值得推广。中央广播电视总台提出“总台算法”的模式,在算法推荐机制中融入主流舆论引导的正向因子。人民日报社也成立智慧媒体研究院,推出主流价值“算法”,以主流价值影响力拓展内容风控等业务。

(三) 网民行为自觉: 社会个体推进“信息茧房”治理深入

作为网民的社会公众,应该不断提高媒介素养和网络意识,发挥主体性,规避“信息茧房”的负面效应。在数字时代,移动互联网深度介入社会民众的生活,生活方式的深度媒介化成为趋势,每个网民在日常的信息资讯获取、生活应用使用、社会交往开展,以及出行、娱乐乃至政务诉求都依靠移动端应用或互联网平台,网民每天都在生产“数据”,这些“数据”被互联网平台采集、归类、分析,并以个性化、贴近性的形式间隔性地再次呈现在网民面前。由于算法带来的“信息茧房”精准匹配了个体的信息、娱乐、生活、消费等多项服务,“投其所好”“不断满足”成为主要特征。面对隐私泄露、网络攻击、网络诈骗等负面效应的产生,作为社会个体的网民需要切实提高数字时代的媒介素养,培养科学的网络使用习惯,正确分辨网络信息,规避过度娱乐化、沉溺化的网络娱乐应用,提高网民群体的媒介接触与自我约束能力。特别是青少年网民群体、老年网民群体,需要正确认知互联网平台的信息推送与网络应用服务,避免因算法推荐服务产生网

络沉迷、网络打赏等过度使用行为，并不断提高 惯、思维、行为等多维视角出发，强化主动性“破现实社会交往的能力。社会个体需要从自身的习 茧”，为网络空间协同治理发挥自身作用。

六、模型的综合评价与改进

6.1 模型的优点

(1) **问题一**采用 LDA 与 BiLSTM 相结合的建模方法，相较传统的时间序列分析，LDA-BiLSTM 模型在主题抽取，情感倾向分析方面精度大大提高，使得模型更加的客观真实；

(2) **问题二**采用 SMPN-互联网情绪传播模型对中立观点与观点极化现象的产生机制进行了细致分析，相较于传统的 SIR 模型，本文构建的模型更符合互联网舆情传播现状，可操作性强。采用网络演化模型对“回声室效应”、“尖叫效应”与“信息茧房”困境的形成机理进行了动态分析，尽量避免主观因素对形成机制的影响。并且，本文利用 AHP 对 5 个指标赋予权重，指标的选取全面考虑了已知信息与题设要求，评价模型可靠性强；

(3) **问题三**在问题二舆情演化影响因素评价模型的基础上，针对具有突出影响的重要指标提出相应的应对策略，相应的策略与政策建议具有较强的可行性。

6.2 模型的缺点

(1) **问题一**出于对实验结果的考量，对舆情演化分析的部分语料文本进行了删除，忽略了这部分数据可能带来的影响，不够全面；

(2) **问题二**中针对舆情演化影响因素的评价采用的是层次分析法，未能采用相关评价模型对影响因素指标的设定进行检验。

6.3 模型的改进

(1) 对舆情传播影响因素的评价可以考虑采用其余的评价模型；

(2) 在对于中立观点与观点极化现象的机制分析中，可以考虑其他的约束条件与影响因素，充分考虑舆情传播演化过程中的现实条件，使模型更加贴合实际。

参考文献

- [1] 周辰璐. 微博热搜对“信息茧房”的强化效果研究 [D]. 黑龙江大学,2020.DOI:10.27123/d.cnki.ghlju.2020.000889.
- [2] 叶艳, 吴鹏, 周知, 黄伟, 张莉曼. 基于 LDA-BiLSTM 模型的在线医疗服务质量识别研究 [J]. 情报理论与实践,2022,45(08):178-183+168.DOI:10.16353/j.cnki.1000-7490.2022.08.024.
- [3] 王桂兰. 在线社交网络网民群体情绪传播建模研究 [D]. 重庆理工大学,2022.DOI:10.27753/d.cnki.gcqgx.2022.000101.
- [4] 刘耀焯, 马书琴, 赵丹. 新媒体时代短视频网络舆情传播的影响因素及动态演化研究 [J]. 情报科学,2022,40(04):179-185.DOI:10.13833/j.issn.1007-7634.2022.04.022.

附录

问题一困惑度计算代码:

```
1  # 准备数据
2  PATH = "abortion_4000.txt"
3
4  file_object2 = open(PATH, encoding='utf-8', errors='ignore').read().split('\n')
5  data_set = [] # 建立存储分词的列表
6  for i in range(len(file_object2)):
7      result = []
8      seg_list = file_object2[i].split() # 读取每一行文本
9      for w in seg_list: # 读取每一行分词
10         result.append(w)
11     data_set.append(result)
12     # print(data_set) # 输出所有分词列表
13
14     dictionary = corpora.Dictionary(data_set) # 构建 document-term matrix
15     corpus = [dictionary.doc2bow(text) for text in data_set]
16     Lda = gensim.models.ldamodel.LdaModel # 创建LDA对象
17
18
19     # 计算困惑度
20     def perplexity(num_topics):
21         ldamodel = Lda(corpus, num_topics=num_topics, id2word=dictionary, passes=50) #
22             passes为迭代次数
23         ldamodel = Lda(corpus, num_topics=num_topics, id2word=dictionary, passes=50) #
24             passes为迭代次数
25         # print(ldamodel.print_topics(num_topics=num_topics, num_words=7)) #
26             num_words为每个主题下的词语数量
27         # print(ldamodel.log_perplexity(corpus))
28         return ldamodel.log_perplexity(corpus)
29
30
31     # 绘制困惑度折线图
32     x = range(1, 9) # 主题范围数量
33     y = [perplexity(i) for i in x]
34     plt.plot(x, y)
35     plt.xlabel('主题数目')
36     plt.ylabel('困惑度大小')
37     plt.rcParams['font.sans-serif'] = ['SimHei']
38     matplotlib.rcParams['axes.unicode_minus'] = False
39     plt.title('主题-困惑度变化情况')
40     plt.show()
```

问题一 BiLSTM 情感分析代码：

```
1     txtpath = "gun_control.txt"
2
3     # 这里就是预测完输出的文件，就是把结果输出到哪里，文件名字自己随意修改。
4     outtxtpath = 'output_gun_control.xlsx'
5
6     # 将数据集文件转换为list格式
7     with open(txtpath, 'r', encoding='utf-8') as f:
8         test_text = []
9         for line in f:
10             test_text.append(line.strip())
11         print(test_text)
12
13     # 加载预训练模型
14     senta = hub.Module(name="senta_bilstm")
15
16     # 将得到的数据list进行预测
17     input_dict = {"text": test_text}
18     results = senta.sentiment_classify(data=input_dict)
19
20     # 将输出结果写入文件
21     workbook = xlswriter.Workbook(outtxtpath) # 建立文件
22     worksheet = workbook.add_worksheet()
23     id = 0
24     for result in results:
25         print(result['text'] + '\t' + result['sentiment_key'])
26         worksheet.write(id,0,result['text'])
27         worksheet.write(id,1,result['sentiment_key'])
28         id+=1
29     workbook.close()
30     print('结果已经输入到' + outtxtpath + '文件')
```

问题一 LDA 模型代码：

```
1     def stopwordslist(filepath):
2         stopwords = [line.strip() for line in open(filepath, 'r', encoding='utf-8').readlines()]
3         return stopwords
4
5     # 对句子进行分词
6     def seg_sentence(sentence):
7         sentence = re.sub(u'[0-9\.\.]+', u'', sentence)
8         sentence_segged = jb.cut(sentence.strip())
9         stopwords = stopwordslist('stopwords.txt') # 这里加载停用词的路径
```



```

10     outstr = ''
11     for word in sentence_seg:
12         if word not in stopwords and word.__len__()>1:
13             if word != '\t':
14                 outstr += word
15                 outstr += " "
16     return outstr
17
18
19     inputs = open('gun_control.txt', 'r', encoding='utf-8')
20
21     outputs = open('gun_control_out.txt', 'w', encoding='utf-8')
22     for line in inputs:
23         line_seg = seg_sentence(line) # 这里的返回值是字符串
24         outputs.write(line_seg + '\n')
25     outputs.close()
26     inputs.close()
27
28     train = []
29
30     fp = open('gun_control_out.txt', 'r', encoding='utf8')
31     for line in fp:
32         if line != '':
33             line = line.split()
34             train.append([w for w in line])
35
36     dictionary = corpora.Dictionary(train)
37
38     corpus = [dictionary.doc2bow(text) for text in train]
39
40     lda = LdaModel(corpus=corpus, id2word=dictionary, num_topics=5, passes=100)
41     # num_topics: 主题数目
42     # passes: 训练轮次
43     # num_words: 每个主题下输出的term的数目
44
45     for topic in lda.print_topics(num_words = 5):
46         termNumber = topic[0]
47         print(topic[0], ':', sep='')
48         listOfTerms = topic[1].split('+')
49         for term in listOfTerms:
50             listItems = term.split('*')
51             print(' ', listItems[1], '(', listItems[0], ')', sep='')
52
53     d=pyLDAvis.gensim.prepare(lda, corpus, dictionary)
54     # pyLDAvis.show(d)
55     pyLDAvis.save_json(d, 'lda.json')

```

```

56
57 # Read JSON file
58
59 import json
60
61 with open('lda.json', 'r') as myfile:
62     data = myfile.read()
63
64     json_data = json.loads(data)
65
66 # Plot with matplotlib
67 pyLDAvis.save_json(d, 'lda.json')
68
69 import matplotlib.pyplot as plt
70
71 x_max = max(json_data['mdsDat']['x']) + (max(json_data['mdsDat']['x']) -
72     min(json_data['mdsDat']['x']))
73 y_max = max(json_data['mdsDat']['y']) + (max(json_data['mdsDat']['y']) -
74     min(json_data['mdsDat']['y']))
75 x_min = min(json_data['mdsDat']['x']) - (max(json_data['mdsDat']['x']) -
76     min(json_data['mdsDat']['x']))
77 y_min = min(json_data['mdsDat']['y']) - (max(json_data['mdsDat']['y']) -
78     min(json_data['mdsDat']['y']))
79
80 plt.axis([x_min, x_max, y_min, y_max])
81
82 # Depending on the number of topics, you may need to tweak the paremeters (e.g. the size
83     of circles be Freq/100 or Freq/200, etc)
84
85 for i in range(len(json_data['mdsDat']['x'])):
86     circle = plt.Circle((json_data['mdsDat']['x'][i], json_data['mdsDat']['y'][i]),
87         radius=json_data['mdsDat']['Freq'][i] / 100)
88     plt.gca().add_artist(circle)
89
90 plt.show()

```

问题一词频分析代码:

```

1 plt.rcParams['font.sans-serif'] = [u'SimHei']
2 plt.rcParams['axes.unicode_minus'] = False
3
4 def stopwordslist():
5     stopwords = [line.strip() for line in open('stopwords.txt', encoding='UTF-8').readlines()]
6     return stopwords
7

```

```

8     def get_plt(data, title):
9         x = [i[0] for i in data]
10        y = [i[1] for i in data]
11        fig, ax = plt.subplots()
12        ax.barh(range(len(x)), y, color='blue')
13        ax.set_yticks(range(len(x)))
14        ax.set_yticklabels(x)
15        plt.rcParams['font.sans-serif'] = [u'SimHei']
16        plt.rcParams['axes.unicode_minus'] = False
17        plt.title(title, fontsize=10)
18        plt.ylabel("词")
19        plt.xlabel("次数")
20        plt.show()
21
22        words_str = ""
23        with open("gun_control.txt", 'r', encoding='utf-8') as f:
24            for line in f:
25                line =
                    re.sub(u"[0-9\s+./,%^*()?;,:- 【】 +\"\\' ]+|[+——! , ;:。 ? 、~@# $ % …… &* ( ) > < -]+",
                        "", line) # 去掉多余字符
26                if line == "":continue
27                line = line.replace("\n", "") # 去掉换行符
28                seg_list = jieba.cut(line, cut_all=False)
29                words_str += (" ".join(seg_list))
30
31                stopwords = stopwordslist()
32                words = [word for word in words_str.split(" ") if word not in stopwords and len(word) > 1]
33
34                word_counts = Counter() # 词频统计
35                for x in words:
36                    word_counts[x] += 1
37                get_plt(word_counts.most_common(20), "词频统计top20") #生成统计图

```

问题二 SMPN-互联网情绪传播模型代码:

```

1     max_iter_num = 5 # 模拟的次数
2     G = nx.karate_club_graph() # 空手道俱乐部
3
4     for edge in G.edges:
5         G.add_edge(edge[0], edge[1], weight=random.uniform(0, 1)) # 可不可以作为权值 病毒的感染能力
6         for node in G:
7             G.add_node(node, state=0) # 用state标识状态 state=0 未激活, state=1 激活
8
9         seed = 1 # 选定33作为传染源
10        G._node[seed]['state'] = 1 # 表示33是感染的

```

```

11
12 all_infect_nodes = [] # 所有被感染的节点放在这里
13 all_infect_nodes.append(seed)
14
15 infected_graph = nx.Graph() # 被激活的图
16 infected_graph.add_node(seed)
17
18 for i in range(max_iter_num):
19     new_infect = list() # 新被感染的
20     # t1 = '%s time' % i + ' %s nodes' % len(all_infect_nodes)
21     # print(t1) # 当前有多少个节点被感染
22     #
23     # # 画图
24     # plt.title(t1)
25     nx.draw(infected_graph, with_labels=True)
26     plt.show()
27
28     # 感染的机会不止一次
29     for v in all_infect_nodes:
30         for nbr in G.neighbors(v):
31             if G._node[nbr]['state'] == 0: # 如果这个邻居节点没被感染
32                 edge_data = G.get_edge_data(v, nbr)
33                 if random.uniform(0, 1) < edge_data['weight']:
34                     G._node[nbr]['state'] = 1
35                     new_infect.append(nbr)
36                     infected_graph.add_edge(v, nbr) # 画图 添加边
37
38     all_infect_nodes.extend(new_infect) # 将新感染的添加到
39     print('all_active_nodes:', all_infect_nodes)

```

问题二與情演化影响因素评价模型代码:

```

1     def __init__(self, criteria, b):
2         self.RI = (0, 0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45, 1.49)
3         self.criteria = criteria
4         self.b = b
5         self.num_criteria = criteria.shape[0]
6         self.num_project = b[0].shape[0]
7
8         def cal_weights(self, input_matrix):
9             input_matrix = np.array(input_matrix)
10            n, n1 = input_matrix.shape
11            assert n == n1, '不是一个方阵'
12            for i in range(n):
13                for j in range(n):

```

```

14     if np.abs(input_matrix[i, j] * input_matrix[j, i] - 1) > 1e-7:
15         raise ValueError('不是反互对称矩阵')
16
17     eigenvalues, eigenvectors = np.linalg.eig(input_matrix)
18
19     max_idx = np.argmax(eigenvalues)
20     max_eigen = eigenvalues[max_idx].real
21     eigen = eigenvectors[:, max_idx].real
22     eigen = eigen / eigen.sum()
23
24     if n > 9:
25         CR = None
26         warnings.warn('无法判断一致性')
27     else:
28         CI = (max_eigen - n) / (n - 1)
29         CR = CI / self.RI[n]
30     return max_eigen, CR, eigen
31
32     def run(self):
33         max_eigen, CR, criteria_eigen = self.cal_weights(self.criteria)
34         print('准则层：最大特征值{:<5f},CR={:<5f},检验{}通过'.format(max_eigen, CR, '' if CR < 0.1
35             else '不'))
36         print('准则层权重={}\n'.format(criteria_eigen))
37
38         max_eigen_list, CR_list, eigen_list = [], [], []
39         for i in self.b:
40             max_eigen, CR, eigen = self.cal_weights(i)
41             max_eigen_list.append(max_eigen)
42             CR_list.append(CR)
43             eigen_list.append(eigen)
44
45         pd_print = pd.DataFrame(eigen_list,
46             index=['准则' + str(i) for i in range(self.num_criteria)],
47             columns=['方案' + str(i) for i in range(self.num_project)],
48             )
49         pd_print.loc[:, '最大特征值'] = max_eigen_list
50         pd_print.loc[:, 'CR'] = CR_list
51         pd_print.loc[:, '一致性检验'] = pd_print.loc[:, 'CR'] < 0.1
52         print('方案层')
53         print(pd_print)
54
55         # 目标层
56         obj = np.dot(criteria_eigen.reshape(1, -1), np.array(eigen_list))
57         print('\n目标层', obj)
58         print('最优选择是方案{}'.format(np.argmax(obj)))
59         return obj

```

```

59
60
61 if __name__ == '__main__':
62     # 准则重要性矩阵
63     criteria = np.array([[1, 2, 7, 5, 5],
64         [1 / 2, 1, 4, 3, 3],
65         [1 / 7, 1 / 4, 1, 1 / 2, 1 / 3],
66         [1 / 5, 1 / 3, 2, 1, 1],
67         [1 / 5, 1 / 3, 3, 1, 1]])
68
69     # 对每个准则，方案优劣排序
70     b1 = np.array([[1, 1 / 3, 1 / 8], [3, 1, 1 / 3], [8, 3, 1]])
71     b2 = np.array([[1, 2, 5], [1 / 2, 1, 2], [1 / 5, 1 / 2, 1]])
72     b3 = np.array([[1, 1, 3], [1, 1, 3], [1 / 3, 1 / 3, 1]])
73     b4 = np.array([[1, 3, 4], [1 / 3, 1, 1], [1 / 4, 1, 1]])
74     b5 = np.array([[1, 4, 1 / 2], [1 / 4, 1, 1 / 4], [2, 4, 1]])
75
76     b = [b1, b2, b3, b4, b5]
77     a = AHP(criteria, b).run()

```
