基于流行度的在线社交网络新闻传播分析方法研究

A Study on News Propagation Analysis Method for Online Social Network Based on Popularity

一　级　学　科：电子信息

研究方向（领域）：计算机技术

作　者　姓　名：刘兴宇

指　导　教　师：张怡 副教授

企　业　导　师：张文博 正高级工程师

答辩日期	2024 年 5 月 25 日		
答辩委员会	姓名	职称	工作单位
主席	毕重科	教授	天津大学
委员	汤善江	副教授	天津大学
	张萌	高级工程师	天津瑞发科半导体技术有限公司

天津大学智能与计算学部

二〇二四年六月

独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 **天津大学** 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名:　　签字日期:　　年　　月　　日

学位论文版权使用授权书

本学位论文作者完全了解 **天津大学** 有关保留、使用学位论文的规定。特授权 **天津大学** 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名:　　导师签名:

签字日期:　　年　　月　　日　签字日期:　　年　　月　　日

摘　　要

随着在线社交网络的迅速发展，分析在线社交网络上的新闻传播成为近年来研究的热点内容。其中，新闻流行度是研究者们关注的重点方向。然而，现有的新闻流行度预测方法中大多仅关注网络结构、时间等特征，忽视了用户的主观传播意愿，影响了预测的准确度。同时，新闻流行度指标无法直观地展现新闻传播的动态过程，缺少相关的细节信息，不能满足研究人员的分析需要。

针对现有的流行度预测方法中忽略用户主观传播意愿的问题，本文提出了一个基于传播意愿的在线社交网络新闻流行度预测模型 WillCas。WillCas 模型通过计算用户活跃程度、前驱用户影响力和基于用户画像的信任程度来量化地表示用户传播意愿，并从用户传播意愿、时间特征、网络结构特征三个方面进行分析，计算新闻的流行度。本文使用一个注意力图神经网络实现了 WillCas 模型，并在两个真实数据集上进行了对比实验和消融实验。对比实验的结果表明，相比于基准模型，WillCas 模型具有更好的准确性。消融实验证明了 WillCas 模型考虑的各部分特征的有效性。

同时，为了在预测新闻流行度的基础上更直观地展现新闻传播的动态过程，便于研究人员进行分析，本文提出了一种基于流行度的在线社交网络新闻传播过程分析方法 SimCas。SimCas 方法设计了一种带流行度约束的新闻传播过程模拟算法 PopSim，使用 WillCas 模型预测的新闻流行度约束模拟过程，生成一种传播网络，并计算其生成概率。SimCas 方法使用 PopSim 算法进行多次模拟，并设计了一个概率加权新闻传播网络生成算法 ProNet，将多次模拟结果聚合为一个概率加权新闻传播网络。之后，SimCas 方法根据模拟结果，使用一个可视分析系统 SimVis 对新闻传播过程进行分析。本文通过真实数据集上的实验验证了 SimCas 方法的模拟结果具有较好的准确性，并通过案例分析的方式进一步评估了其分析功能的有效性。

关键词：在线社交网络，流行度预测，传播过程分析

ABSTRACT

With the rapid development of online social networks, analyzing news propagation on these platforms has become a hot topic in recent research. News' popularity is one of the key focus for researchers. However, most existing methods for predicting news' popularity only consider features such as network structure and time, while ignoring users' subjective willingness of propagation, which affects the accuracy of predictions. At the same time, news' popularity cannot intuitively show the dynamic propagation process of news, which lacks relevant detailed information and cannot meet the analytical needs of researchers.

In response to the issue of existing popularity prediction methods neglecting user's subjective willingness, this paper proposes a model for predicting news' popularity on online social networks based on the willingness of propagation, WillCas. WillCas quantitatively represents users' willingness of propagation by calculating user's activity level, the influence of precursor users, and the trust based on user features. The model analyze from three perspectives: users' willingness of propagation, temporal features, and network structural features, to caculate the popularity of news. This paper use a attention-based graph neural network to implement the WillCas model. And verify this model through comparative and ablation experiments on two real-world datasets. The results of the comparative experiments indicate that the WillCas model has a better accuracy compared to baseline models. The ablation experiments confirm the effectiveness of the features considered by the WillCas model.

Meanwhile, in order to intuitively demonstrate the news propagation process based on the prediction of news popularity and help researchers' analysis, this paper proposes a popularity-based propagation process analyze method of news on online social networks, SimCas. The SimCas method designs a news propagation process simulation algorithm with popularity constraints, PopSim. This algorithm uses the predicted popularity of news by WillCas model to constrain the simulation process, generates a propagation network, and calculates its generation probability. The SimCas method uses the PopSim algorithm to conduct multiple simulations and designs a probability-weighted news propagation network generation algorithm, ProNet, to aggregate multiple simulation results into a probability-weighted news propagation network. After that, the

SimCas method uses a visual analysis system, SimVis, to analyze the news propagation process based on the simulation results. This article verifies the good accuracy of the simulation results of the SimCas method through experiments on real datasets and further evaluates the effectiveness of its analysis function through case analysis.

KEY WORDS： Online Social Networks, Popularity Prediction, Propagation Process Analysis

目　录

第 1 章　绪论

1.1　研究背景与意义

近年来，互联网和移动通信技术的不断发展，深刻地改变了人们获取新闻的方式。特别是在线社交网络的兴起，例如推特（Twitter)、脸书（Facebook）、微博等平台，为新闻的生成和传播带来了前所未有的便利[1]。在线社交网络（Online Social Networks）是一种基于互联网的社交平台，它允许用户生成在线内容，并与其他用户建立社交关系，通过转发、点赞、评论等方式与其他用户就感兴趣的话题进行互动[2]。借助在线社交网络，新闻可以摆脱地理和媒介因素的限制，迅速传播到世界各地的用户之中。在线社交网络以其独特的低成本、时效性和便捷性的特点，使得越来越多的人更加习惯于通过在线社交网络获取新闻信息。例如，美国皮尤研究中心的调查显示，2020 年约有 79% 的美国成年人使用在线社交网络获取新闻[3]。

另一方面，在线社交网络低成本和便捷性的特点同样导致其中包含的信息量十分巨大，在推特平台上，全球每分钟都会新增超过 30 万条推文[4]。然而，用户的关注度却是有限的，在有限的时间内不可能关注到所有的信息。在每天产生的海量信息中，大部分用户的关注度只集中于很少的一部分信息上，呈现出明显的长尾效应[5]，即少部分信息获得了大量关注度，而大部分信息只获得很少的关注度。例如，在微信公众平台上，每天会产生大约 150 万篇新文章，其中只有 0.07% 的文章被分享超过 1 万次[6]。因此，对在线社交网络的新闻传播进行分析，判断哪些新闻更有竞争力，分析其传播过程和受欢迎原因，对于新闻媒体、平台运营者和政策制定者都具有十分重要的意义。而预测流行度则是分析新闻传播的一种重要方法。

在线社交网络的新闻流行度预测在许多现实领域都体现出了丰富的研究价值。在商业领域，一方面，预测新闻未来可能的流行度可以让平台及时发现有广泛传播潜力的新闻，提前倾斜平台资源，吸引大量用户关注。另一方面，通过分析不同新闻在用户中的流行度，可以优化平台的个性化推荐系统，改进广告投放策略，从而更加贴合用户喜好，提供更好的用户体验，获取更多利润。在管理领域，平台可以借助新闻流行度的变化趋势，甄别虚假信息，及时遏制谣言和假新闻的扩散[7]。在决策领域，具有较高流行度的新闻内容反映了社会关心的热点话题，不仅能够为决策者提供建议，而且能对自然灾害和突发事件做出及时响应，

还可以对潜在威胁进行预警。

预测在线社交网络上的新闻流行度实际上是一个信息级联预测问题。在线社交网络上的新闻传播本质上是一种信息级联（Information Cascades），它建立在用户网络之上，遵循“发布者-接收者”模式[8]。在在线社交网络中，用户之间可以相互建立关注关系，从而形成用户网络。这种关系是单向的，一名用户可以通过关注另一名用户的方式，成为其“粉丝”，接收到其发布的内容，并通过“转发”操作将该内容传播给他的粉丝[9]。

但是，目前的流行度预测方法大多关注于时间、文本内容、网络结构等客观特征，忽视了作为传播主体的用户的主观意愿，影响了预测的准确度。由于个人性格等因素的影响，用户在社交网络上传播信息的活跃程度是不同的，不同用户在用户网络中的影响力也是不同的。同时，回音室效应表明，用户更有可能接受来自他所信任的用户的信息[10]，因而也会影响他们对信息的传播意愿。因此，需要在预测过程中考虑包括用户活跃度、前驱用户影响力、信任值在内的用户主观传播意愿。

另一方面，流行度只能体现新闻传播的总体规模，忽略了传播过程中的具体细节，其形式不够直观，不能满足研究人员对传播过程、用户特征的分析需求。因此需要设计一种准确、直观的在线社交网络新闻传播过程分析方法，为研究人员的分析提供帮助。

为了解决上述问题，本文提出了一个基于传播意愿的在线社交网络新闻流行度预测模型 WillCas，从用户活跃程度、前驱用户影响力、基于用户画像的信任程度三个方面量化用户的传播意愿，并综合考虑时间特征和网络结构特征，预测新闻流行度。本文构建了一个图神经网络框架来实现 WillCas 模型，并采用注意力机制对模型进行优化。在此基础上，本文还提出了一种基于流行度的在线社交网络新闻传播过程分析方法 SimCas。SimCas 方法设计了一个带流行度约束的新闻传播过程模拟算法 PopSim，利用流行度预测结果约束模拟过程，得到一次模拟的新闻传播网络及其生成概率。同时设计了一个概率加权新闻传播网络生成算法 ProNet，将多次模拟结果聚合为一个概率加权新闻传播网络，便于后续分析。根据模拟结果，SimCas 方法通过一个可视分析系统 SimVis 完成分析工作。

本文的研究仅考虑公开的在线社交网络平台（如微博、推特等），对于私域在线社交网络平台（如微信），由于用户之间的关注关系不是透明的，用户无法主动发现未关注的用户，并且难以获取到完整的传播级联数据，因此不在本文的研究范围之内。

1.2 国内外研究现状

近年来，在线社交网络成为学者们关注的热门领域。研究人员提出了许多优秀的级联预测模型和算法，并将其应用到在线社交网络的新闻流行度预测工作中。一些研究则关注于在线社交网络的用户关系，通过多种方法评估用户之间的信任程度。还有许多学者设计了创造性的可视化视图，对在线社交网络的新闻传播进行可视分析。本小节从级联预测、用户信任预测、新闻传播可视分析三个方面，对当前领域国内外的研究现状进行分类介绍。

1.2.1 级联预测

根据 1.1 章节中的介绍，在线社交网络上的新闻流行度预测本质上是一个级联预测问题。因此，本小节将首先对级联预测工作进行分类，再根据使用的研究方法分别介绍相关的研究工作。

1.2.1.1 级联预测的分类

现有的信息级联预测研究可以从任务粒度、预测时间、预测目标、研究方法四个角度进行分类。

按任务粒度分，信息级联预测可以分为宏观预测、微观预测和中层预测。其中，宏观预测主要根据级联的整体特征，从宏观层面对级联的整体行为进行建模，其研究目的是预测级联未来的大小。例如，Xiao 等人[11] 通过分析级联的时间特征，预测未来一段时间内的级联增长大小；Andery 等人[12] 根据推文的内容特征，预测推文的最终传播规模。对于微观预测，则更加关注用户个体的行为，从微观尺度对级联的传播过程进行建模。例如，Yang 等人[13] 采用深度学习模型对用户传播概率进行建模，建立了一种级联扩散模型 NDM。Wang 等人[14] 提出了一种基于拓扑结构的循环神经网络模型，对级联扩散的范围和扩散时间进行建模。而中层预测则侧重于提取用户社区，分析用户群体行为，探究它们对级联传播产生的影响。Weng 等人[15] 分析了社区结构的特点，说明了社区结构对信息传播的影响。Mcauley 等人[16] 提出了一种基于图结构的用户社区检测算法，用于发现在线社交网络中的用户社交圈。

按预测时间分，信息级联预测包括事前预测和事后预测。事前预测指在级联生成之前，根据文本内容、发布者用户信息等已知特征，预测级联的未来大小和传播过程。对于广告投放等领域来说，事前预测可以在内容发布前估计其传播效果，可以帮助内容发布者最大程度地降低风险、增加利润。但事前预测中可以依赖的特征数量较少，并且在现实世界中常由于平台政策、隐私敏感等原因难以获取，目前事前预测仍然面临较大挑战。事后预测则指的是在级联生成后的一段时

间，根据已经观察到的级联特征进行预测。根据 Gabor 等人[17] 的发现，级联早期的特征与未来特征之间存在很强的相关性，在早期阶段更受欢迎的信息往往最终会更加流行。因此，根据级联早期特征进行事后预测是目前主流的研究方向。例如，Peng 等人[18]、Cheng 等人[19] 和 Zaman 等人[20] 的工作中都使用了级联的早期特征估计级联的未来大小。

按预测目标分，信息级联预测有分类预测和回归预测两种目标。早期的研究工作中，通常将级联预测视为一个分类问题。在给定一个阈值或几个区间的前提下，使用机器学习的方法，将预测级联最终大小转化为一个分类任务。例如 Peng 等人[21] 通过机器学习的分类方法判断级联是否会成为爆发状态，Gao 等人[22] 使用多种分类器预测级联最终大小会落在哪个给定区间中。分类预测虽然相对容易且准确度较高，但估计值过于粗略，难以满足精确的研究需求。近年来的工作更加关注于回归预测，即预测级联未来能达到的确切流行度值。例如 Cao 等人[1]、Chen 等人[2] 和 Zhong 等人[5] 的工作都将预测级联在未来一段时间的增量值作为预测目标。

按研究方法分，现有的级联预测主要采用了三种研究方法：基于特征工程的方法、基于过程模型的方法和基于深度学习的方法。在接下来的三个小节中，将分别详细介绍与这三种研究方法相关的工作。

1.2.1.2 基于特征工程的级联预测

早期的信息级联预测研究主要关注于级联本身的各种特征，包括时间、用户信息、网络结构、文本内容等特征，使用传统的特征工程方法提取多种特征，并使用如支持向量机、随机森林等经典机器学习的方法建立预测模型。

时间特征是影响级联传播的关键因素之一。现有的工作主要关注发布时间、响应时间等特征。Wu 等人[23] 利用初始信息的发布时间，采用指数回归预测模型预测级联的传播规模。Sasa 等人[24] 设计了多个局部模型，使用一天中不同发布时间的样本分别进行训练，以改进预测效果。在 Zaman 等人[20] 的工作中，综合考虑了发布时间和用户平均响应时间，采用朴素贝叶斯模型进行预测。尽管时间特征十分重要，但其与上下文拥有强相关性，在不同条件下差异较大。因此总结出泛化性强的时间影响函数仍是一项挑战。

参与级联的用户携带的大量个人信息反映了用户的内在特征，因此也得到了许多研究者的重点关注。其中最受欢迎的是用户的关注者数量，它最能体现用户的影响力[20]。例如，Suh 等人[25] 分析了用户关注者数量和关注者年龄，使用了包括隐马尔可夫模型在内的多种机器学习模型进行预测。Maximilian 等人[26] 同样采用了用户关注者数量衡量用户影响力，判断推文是否会“病毒式传播”。另外，用户的账号资料、历史行为同样也被用于表示用户特征，如 Tatar 等人[27] 考

虑了发布者地理位置、年龄等人因素，Hong 等人[28] 则分析了用户历史点赞、评论行为的影响。然而受限于平台的隐私政策和用户个人的隐私可见性，部分用户信息难以获取，这限制了对用户信息的分析能力。

网络结构反映了用户关系和级联传播过程，是影响级联传播的另一个重要因素。研究者们通常将用户网络和级联用有向图的形式表示，并使用图结构特征进行分析。Peng 等人[18] 计算了早期微博网络的边缘密度与深度，分析了这两个结构特征与级联最终大小的相关性。Sharad 等人[29] 综合考虑了网络拓扑结构和其他特征，提出了一种病毒结构性传播模型，用于判断级联的传播模式。Salman 等人[30] 基于用户之间的互动行为，如转发、评论、点赞等，构建了一个交互网络，用于分析用户间的交互模式。Dong 等人[31] 同样根据用户之间的提及关系（用户可以使用 @ 功能在自己的内容中提及另一用户）构建了类似的交互网络。

此外，基于信息文本内容的特征也是研究者们关注的热点之一。通过分析文本语义，可以提取主题信息和情绪信息。例如，Hong 等人[28] 使用了词频-逆文档频率模型（TF-IDF）和隐狄利克雷分布模型（LDA）来提取文本的主题分布。Bandari 等人[32] 进一步考虑了主题类别、语言的主体性和命名实体数量等特征。Elham 等人[33] 则分析了包括评论长度、动词/名词数量、可读性在内的语义统计特征。在 Yuan 等人[34] 的工作中，根据语义评估了用户情绪和自我分享意愿，分析了用户情绪对传播的影响。

与其他方法相比，基于特征工程的预测方法通常具有较好的可解释性。但该方法依赖于手工制作的特征工程，不仅需要掌握相关领域的专业知识，还需要耗费大量的人力和计算资源。一些特征由于平台限制和隐私政策等原因难以获取，也限制了特征的选择范围。此外，基于特征的方法依赖于特定的应用场景，其可扩展性和泛化性相对较差。

1.2.1.3 基于过程模型的级联预测

基于过程模型的方法同样是早期的级联预测工作中较为流行的方法。级联的传播可以表示为事件序列，许多研究者以已有的传播动力学模型为基础，对级联的传播过程进行建模。常见的过程模型可以分为泊松点过程模型、霍克斯过程模型、传染病模型三类。

点过程模型是一种描述随机点在空间中随机分布的模型，它通常用于建模随机事件序列，适合用于级联预测中。泊松点过程模型是较为常用的点过程模型，许多工作使用了泊松点过程模型进行预测。例如，Shen 等人[35] 提出了一种基于增强泊松过程的生成概率模型，量化了级联内容对用户的吸引力，并使用了对数正态分布的时间衰减机制和强化机制。Gao 等人[36] 在增强泊松过程模型的基础上进行了扩展，提出了更适用于微博平台的模型 PETM，使用幂律分布代替对数

正态分布表示时间衰减。Lu 等人[37] 根据用户关系将级联事件列分解为树结构，设计了基于复合泊松分布的 RepostTree 模型。

霍克斯（Hawkes）过程模型也是一种常见的点过程模型，它提出了一种自激励机制用于建模过去事件对未来事件的影响。Amir 等人[38] 使用用户关注者数量衡量自激励强度，并使用帕累托分布作为时间衰减函数。Mishra 等人[39] 定义了一个分支因子用于优化霍克斯模型，分支因子设定了级联预期大小的上限阈值，超过这个阈值大小的级联被视为不可预测的。Ding 等人[40] 则在霍克斯模型的基础上额外考虑了文本内容的情感影响，设计了自我激励和交叉激励两个自激励机制，构建了双重情感霍克斯过程模型 DSHP。

传染病模型原本是应用于流行病学的数学模型，许多研究者将其应用到信息级联传播研究中，将级联的传播类比为流行病传播过程。常见的传染病模型有 SI 模型、SIS 模型、SIR 模型等。Ding 等人[41] 提出了一种基于传染病模型的微博传播模型 SCIR，引入了一种新的节点状态——接触状态，用于表示接收到信息但尚未决定是否传播信息的用户。Zhao 等人[42] 设计了一种结合传染病模型和霍克斯模型的预测模型 SEISMIC，采用当前转发次数、网络用户总数来估计传染性，并拟合一个平均反应时间作为时间衰减函数的初始值，再让其呈现幂律衰减。Li 等人[43] 则改进了 SEISMIC 模型，使用传播速度修正传染性，并使用修正后的传染性及时调整分支因子。

还有一些研究使用了其他种类的过程模型。如 Yu 等人[44] 采用了生存分析模型的思想，使用用户特征和结构特征，将用户的转发事件类比为生存过程。Li 等人[45] 参考了博弈论中有关决策动机的理论，建立了基于用户回报的传播预测模型，从影响力回报和偏好回报两个角度计算用户决策得到的回报，并计算用户被激活的概率。

相比基于特征工程的方法，基于过程模型的方法不需要大量的特征工程，并且由于其对级联的传播过程进行了建模，因此具有良好的可解释性。但这些模型将复杂的级联传播过程进行了简化，因此其预测准确度相对较差，并且对异常值十分敏感[39]。

1.2.1.4 基于深度学习的级联预测

随着深度学习技术的不断发展，越来越多的研究者将深度学习的方法应用到级联预测的研究中。深度神经网络具有强大的学习能力，擅长挖掘数据之间复杂的非线性映射关系，更加适合于复杂的级联传播过程。与基于特征工程和过程模型的方法相比，基于深度学习的方法拥有相对简单的模型结构和更加优秀的预测结果，因而成为近年来最受欢迎的级联预测方法。

Guan 等人[46] 使用卷积神经网络（CNN）对新闻文本内容进行词嵌入，并

使用长短期记忆网络（LSTM）对新闻流行度进行分类预测。Saeed 等人[47] 分别构建基于新闻文本内容、发布时间、传播体量的特征向量，并同时使用 CNN、深度神经网络（DNN）、LSTM 的方法学习特征表示，预测流行度结果。Dou 等人[48] 将新闻文本内容、用户互动数据、发布时间等特征定义为内容实体（Content Entities），并提出了基于 LSTM 的预测模型。通过自适应地结合目标实体与已知实体，从而预测具有相似实体的新闻流行度。Cao 等人[1] 将深度学习与霍克斯过程模型结合起来，建立了 DeepHawkes 模型。将用户信息嵌入为用户向量，作为霍克斯模型中的影响力指标；使用门控循环单元（GRU）对级联传播路径进行编码，用来表示自激励响应；使用非参数的方法，分时间段学习各自的时间衰减函数；最终使用加权和池化层输出级联增量流行度预测结果。

近年来，图表示学习（Graph Embedding）和图神经网络（GNN）技术的发展，使得研究者们可以更好地建模图结构数据，捕获图结构中的复杂模式。因此，许多研究者将图表示学习和图神经网络应用到级联预测工作中。例如，Chen 等人[2] 设计了一种半监督的递归级联卷积神经网络 CasCN，将级联图采样为多个级联子图，采用 LSTM 与图卷积神经网络（GCN）相结合学习子图的结构表示，并使用非参数的方法学习时间衰减函数。Cao 等人[49] 建立了具有两个耦合 GNN 的 Coupled-GNNs 模型，分别用于模拟级联网络上用户影响的传播和用户状态的更新。Wang 等人[50] 综合考虑了网络特征与时间特征，将级联划分为多个快照，并使用基于动态路由的节点表示聚合方法，利用 GCN 学习快照表示，并使用 LSTM 提取时间信息。

深度学习模型的计算复杂度相对较高，因此研究者们使用了多种方法来提升深度学习的模型性能，注意力机制（Attention Mechanis）是其中最常用的方法。注意力机制能够为深度学习模型中不同部分的输入数据赋予不同的权重，让模型在学习时将注意力集中到更重要的部分，从而提升模型性能。Li 等人[7] 建立了一个端到端的图注意力网络 DeepCas，依据深度游走方法（DeepWalk）对级联图进行随机游走采样，并使用双向 GRU 学习节点的隐藏表示，再使用衡量节点影响力的注意力权重进行加权。Zhong 等人[5] 在 DeepCas 的模型框架基础上使用了双层注意力网络，分别用来加权节点的影响力和去除级联序列中的冗余性。Ding 等人[51] 则使用了多头注意力机制来同时考虑文本内容、标题、用户浏览历史的影响。Qi 等人[52] 针对新闻中的内容实体，混合了多头自注意力网络和多头交叉注意力网络，分别用来表示实体间相关性和实体的上下文相关性。Huang 等人[53] 在 Chen 等人[2] 工作的基础上进行了改进，使用图注意力网络和 Transformer 代替 GCN 和 LSTM 学习级联上下文信息。此外，在 Sun 等人[54]、Wang 等人[55] 的工作中也使用了注意力机制。

除注意力机制外，研究者们还使用了其他一些方法来改善性能。如 Wicaksono

等人[56]使用遗传算法来优化超参数的学习，Yang 等人[57]使用强化学习指导微观级联预测，Sanjo 等人[58]使用多模态的方法提升模型性能，等等。

总而言之，相比前两种方法，基于深度学习的方法以其相对简单的模型架构和强大的学习能力在级联预测研究中有着显著的优势, 是目前主流的研究方法。但基于深度学习的方法仍然存在计算成本高、可解释性差的局限性，并且在学习过程中还面临超参数选择、模型调整、过拟合风险等挑战。

1.2.2 用户信任预测

在人际关系中，信任是一个复杂的社会心理概念，Mayer 等人[59]将信任定义为一方因期待另一方将执行对自己重要的特定行为，从而愿意使自己受其影响，无论能否监督或控制另一方。简而言之，信任指的是个体对另一个体或群体可靠性的信心和期望，是一种主观意愿。

在在线社交网络中，用户之间的信任程度是影响他们建立关系、开展互动、分享内容的重要因素之一[60]，用户总是更倾向于从他们更信任的用户处获取新闻。因此，信任是用户传播新闻主观意愿的重要组成部分。信任预测可以定义为预测在线社交网络中可能没有联系的一对用户之间的信任关系的过程[61]。它可以是一个分类问题，即判断用户之间是否互相信任，也可以是一个回归问题，即用量化的形式表示用户之间的信任程度。本小节将详细介绍与用户信任预测相关的研究工作。

Golbeck 等人[62]提出了一种基于“朋友的朋友”（FOAF）概念的信任推测方法，来判断用户网络中哪些用户之间是相互信任的。Tang 等人[63]基于同质性效应，提出相似的用户之间建立信任关系的可能性更高，并通过计算用户在评分网站中评分的余弦相似度来计算用户之间的信任值。Abbasi 等人[64]使用 CredRank 算法评估用户的可信度，通过计算可信度的相似性来判断用户之间的信任情况。Hang 等人[65]提出了一种基于用户网络相似度的预测方法，把信任预测转化为图相似度问题。Adali 等人[66]关注用户之间的对话，根据对话的持续时间和频率预测信任情况。

信任是与上下文相关的，用户之间的信任程度在不同环境中是不同的。Liu 等人[67]提出了高质量信任网络的定义，即更符合上下文情况的信任网络，并且将上下文分为独立上下文和依赖上下文，给出了信任程度的计算公式。其中独立上下文包括用户的社会影响力、偏好、居住地，依赖上下文包括与其他用户之间的关注关系、结点的度。Ghafari 等人[68]利用社会心理学理论启发的社会环境因素，提出了一种基于张量分解的上下文感知信任预测方法 TDTrust，用于预测在特定上下文中用户之间的信任度。

总而言之，在在线社交网络中，大部分用户之间在现实生活中并不相识。因

此许多研究者通过用户信息、历史行为等特征，并结合上下文预测用户之间的信任程度。然而如何综合考虑各方面特征并准确量化信任程度仍然是目前面临的一大挑战。

1.2.3 新闻传播可视分析

可视分析具有易于理解、使用门槛低、便于交互等优点，可以直观地展示在线社交网络上的新闻传播过程，便于研究者深入理解新闻传播的模式。近年来，许多研究者将可视分析的方法应用于与新闻传播相关的工作中，设计了多种优秀的可视化视图。

Chen 等人围绕新浪微博平台上的信息传播过程做了大量研究，在 E-Map 中[69]，设计了一个基于地图隐喻的可视化视图，使用城镇、河流等地图元素对话题随时间的演变过程进行可视化；在 D-Map+ 中[70]，对用户进行聚类分组，划分用户社区，使用六边形区块表示用户，重点研究信息在如何在用户群体之间传播；在 R-Map[71] 中，结合了 E-Map 的地图隐喻和 D-Map+ 中的六边形区块，设计了一个模拟信息传播过程的可视化视图，并重点关注传播中关键人物的作用和话题的转变。除了 Chen 等人的工作外，Cao 等人[72] 设计了一种向日葵隐喻，对不同话题信息的传播过程进行可视化。Ren 等人[73] 使用树状布局展示信息的扩散模式和层次结构，并突出显示传播过程中的关键角色。Liu 等人[74] 建立了一个动态流体模型建模微博的传播过程，并同时分析了微博传播的速率和路径。

总而言之，可视分析的方法可以更加直观地展示新闻传播的过程细节，为研究人员的进一步分析提供帮助。

1.3 本文工作及主要贡献

根据 1.2 章节中的介绍，近年来在线社交网络的新闻流行度预测工作中，更多采用基于深度学习的方法。相比于基于特征工程和过程模型的方法，基于深度学习的方法拥有更好的预测效果。但是，现有的基于深度学习的方法中，普遍只关注时间、文本内容、网络结构等客观特征，忽视了作为传播主体的用户的主观意愿，影响了预测的准确度。同时，流行度仅能反映新闻传播的规模，不能直观地展示新闻传播的具体过程，无法满足分析需求。因此需要设计一种准确、直观的在线社交网络新闻传播过程分析方法，为分析新闻传播提供帮助。

为了解决当前在线社交网络（不包括私域在线社交网络，如微信）新闻流行度预测工作中忽视用户主观传播意愿，导致准确度下降的问题，本文提出了一个基于传播意愿的在线社交网络新闻流行度预测模型 WillCas。同时，为了在预测新闻流行度的基础上更直观地展现新闻传播的动态过程，便于研究人员进

行分析，本文还提出了一种基于流行度的在线社交网络新闻传播过程分析方法SimCas。本文的主要工作和贡献如下：

1. 提出了一个基于传播意愿的在线社交网络新闻流行度预测模型 WillCas。本文从用户活跃程度、前驱用户影响力、基于用户画像的信任程度三个方面量化地计算了用户的传播意愿，并综合考虑了时间特征和网络结构特征，预测新闻的流行度。本文搭建了一个图神经网络框架来实现 WillCas 模型，使用随机游走的方式对传播网络进行采样，采用双向 GRU 学习采样序列的隐藏表示，通过注意力机制加权各类特征，最终输入多层感知机（MLP）得出预测结果。本文在两个真实数据集上进行了对比实验和消融实验。对比实验表明，相比于基准模型，WillCas 模型具有更好的准确性。同时，消融实验证明了 WillCas 模型考虑的各部分特征的有效性。
2. 提出了一种基于流行度的在线社交网络新闻传播过程分析方法 SimCas。SimCas 方法首先对新闻传播过程进行模拟，提出了一种带流行度约束的新闻传播过程模拟算法 PopSim，使用 WillCas 模型预测的流行度约束模拟过程，生成一种新闻传播网络，并计算了生成概率。SimCas 方法使用 PopSim 算法进行多次模拟，并设计了一个概率加权新闻传播网络生成算法 ProNet，将多次模拟结果聚合为一个概率加权新闻传播网络。之后，SimCas 方法根据模拟结果，使用一个可视分析系统 SimVis 对新闻传播过程进行分析。本文通过真实数据集上的实验验证了 SimCas 方法的模拟结果具有较好的准确性，并通过案例分析的方式进一步评估了其分析功能的有效性。

1.4 本文结构

本文的主要结构如图1-1所示。本文共分为五个章节。第一章为绪论，主要介绍了本文的研究背景和意义、相关领域的国内外研究现状以及本文的主要工作和贡献。第二章为相关理论基础，主要介绍了本文涉及到的相关理论和技术方法，主要包括社交网络特性、相关算法、神经网络模型等内容。第三章中详细介绍了本文提出的基于传播意愿的在线社交网络新闻流行度预测模型 WillCas，介绍了其模型的组成，各部分特征的计算方法以及实现方法，并阐述了在真实数据集上的实验过程和实验结果。第四章中详细介绍了本文提出的基于流行度的在线社交网络新闻传播过程分析方法 SimCas，给出了带流行度约束的新闻传播过程模拟算法 PopSim 和概率加权新闻传播网络生成算法 ProNet 的算法流程及伪代码，并介绍了准确性验证的过程和实验结果。此外还介绍了可视化分析系统 SimVis，以及相关的案例分析过程。第五章为总结与展望，回顾了本文的工作并针对工作中

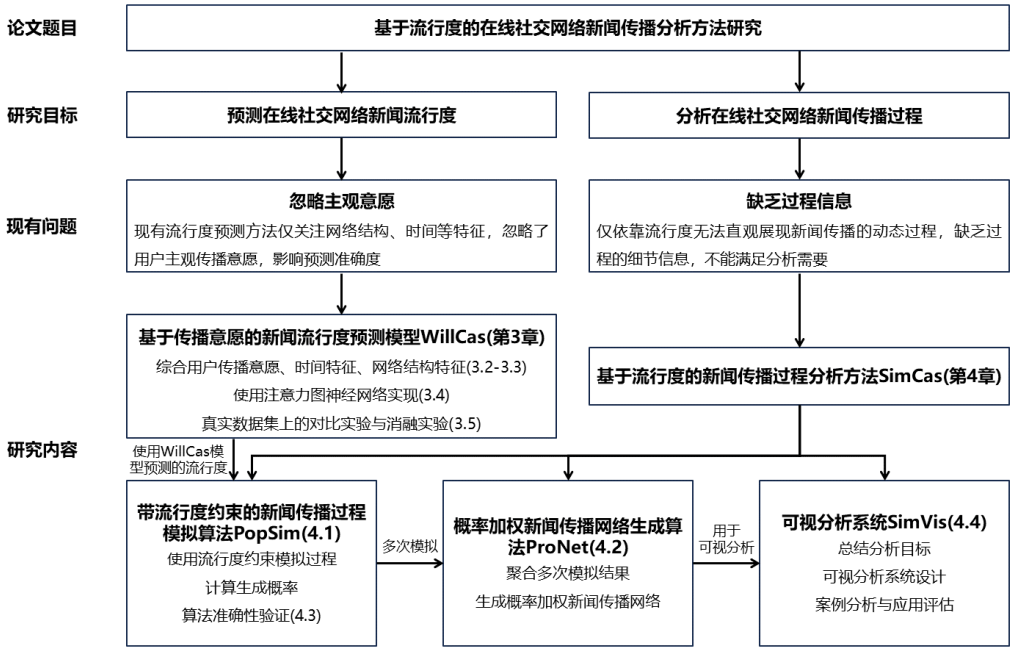


图 1-1　本文结构框架图

有待进一步完善的内容进行了展望。

1.5　本章小结

本章首先介绍了本文的研究背景和意义，包括在线社交网络新闻传播分析的重要性和应用价值、问题的实质以及当前相关研究中存在的问题；之后介绍了级联预测领域、用户信任预测领域和新闻传播可视分析领域的国内外研究现状；最后列出了本文的主要工作和贡献，以及本文的整体结构。

第 2 章　相关理论基础

本章将对本文工作涉及到的相关理论知识和使用的相关技术进行介绍。首先，本章介绍了在线社交网络的定义和特性，之后详细介绍了本文工作中使用到图分析算法和神经网络模型。

2.1　在线社交网络

社交网络（Social Networks）指的是由社会成员互相之间的社交关系而形成的拓扑网络结构，其概念最早由 Barnes 等人[75] 在 1954 年提出。社交网络使得人们可以进行互动并建立新的联系，从而不断扩张自己的社交网络。随着互联网技术和移动通信技术的发展，逐渐出现了大量提供社交网络服务（Social Network Service，简称 SNS）的网站和软件，形成了在线社交网络（Online Social Networks）。

许多研究者都给出了在线社交网络的相关定义，如 Schneider 等人[76] 将在线社交网络定义为具有相同兴趣、活动、背景和友谊的人之间形成的在线社区。大多数在线社交网络都是基于 Web 的，允许用户发布文本、图像、视频等内容，并通过点赞、评论、转发等多种方式与他人进行互动。

在在线社交网络中，一名用户可以将另一名用户加入自己的关注列表，与之建立关注关系，从而接收到对方发布或转发的内容[2]，该用户称为被关注用户的粉丝或追随者（follower）。这种关注关系是单向的，粉丝可以接收到被关注者发布的内容，但被关注者无法接收到粉丝发布的内容，除非被关注者也与粉丝建立关注关系，成为“粉丝的粉丝”，这种情况下通常称两名用户互相关注。因此，对于在线社交网络上的每个用户，都拥有一个关注者用户列表和一个粉丝列表。用户可以从他关注的用户处接收到该用户发布的信息，并通过转发操作将该信息添加到自己发布的内容中，从而将信息发送给他的粉丝，成为信息的传播者。借助转发功能，信息可以在在线社交网络上广泛地传播。

一个在线社交网络可以被形式化定义为一个有向图 $G = (V, E)$，其中 $V = \{v_1, v_2, ..., v_n\}$ 为表示用户的节点集合，$E = \{(v_i, v_j) | v_i, v_j \in V\}$ 为表示用户关注关系的边集合。边 (v_i, v_j) 表示用户 v_i 关注了用户 v_j，即用户 v_i 为用户 v_j 的粉丝。对于任意用户 v_i，其出边邻居节点集合 $Out(v_i)$ 表示用户 v_i 的关注者集合，出度等于关注者数量。同理，其入边邻居节点集合 $In(v_i)$ 表示用户 v_i 的粉丝集合，入

度等于粉丝数量。

2.2 在线社交网络特性

本小节将介绍在线社交网络研究的两个特性：局部聚类系数和小世界网络。

2.2.1 局部聚类系数

聚类系数是用来表示一个图中节点聚集程度的指标，局部聚类系数则是用来表示图中某个节点附近的聚集程度的指标，反映了该节点邻居节点的连通性。局部聚类系数越高，该节点的邻居节点之间的连通性越强[77]。在在线社交网络中，局部聚类系数越高的用户，其相邻用户（关注者和粉丝）之间的联系越紧密。因此局部聚类系数是研究在线社交网络结构的重要特性。有向图的局部聚类系数的计算方法如下：

对于有向图 $G = (V, E)$，其中 $V = \{v_1, v_2, ..., v_n\}$ 为节点集合，$E = \{(v_i, v_j)|v_i, v_j \in V\}$ 为边集合。对于任意节点 v_i，定义其邻居节点（不考虑入边和出边的区别）集合 $N(v_i) = \{v_j|v_j \in V, (v_i, v_j) \in E \vee (v_j, v_i) \in E\}$，则节点 v_i 的局部聚类系数 cn_i 的计算公式为：

$$cn_i = \frac{|\{(v_j, v_k)|j, k \in N(v_i), (v_j, v_k) \in E\}|}{|N(v_i)|(|N(v_i)| - 1)} \tag{2-1}$$

2.2.2 小世界网络

小世界网络的概念由 Watts 等人[77] 在 1998 年提出，它是一种具有较短平均路径长度和较高平均聚类系数的网络。小世界网络被定义为网络中任意两个节点间的平均路径长度 L（节点之间最短路径的平均长度）与网络中节点总数 n 呈对数比例增长，即满足：

$$L \propto logn \tag{2-2}$$

根据定义，小世界网络中的节点之间大多不彼此直接相连，但却具有较高的全局连通性，节点之间可以通过较短的路径相互联系。

在线社交网络是一种典型的小世界网络[78]，在线社交网络中的用户大多没有直接的关注关系，但可以通过较短的传播路径获取到他人发布的信息。小世界网络特性使得信息可以通过用户的转发行为快速传播到在线社交网络的大部分用户中，保证了信息的流通性，简化了传播过程，对研究在线社交网络上的新闻传播具有重要意义。

根据小世界网络的定义，判断一个网络是否具有小世界网络特性通常使用平均路径长度和平均聚类系数两个指标，与等规模（具有相同节点数和边数）的随机网络进行比较。当一个网络满足平均路径长度约等于等规模随机网络的平均路径长度，且平均聚类系数远大于等规模随机网络的平均聚类系数时，该网络即具有小世界网络特性。

具体到计算方法上，对于网络 $G = (V, E)$，$V = \{v_1, v_2, ..., v_n\}$ 为节点集合，有 n 个节点，$E = \{(v_i, v_j)|v_i, v_j \in V\}$ 为边集合。设网络中共有 m 对互相连通的节点，任意一对连通节点 v_{i_1}、v_{i_2} 之间的最短距离为 d_i，则网络 G 的平均路径长度 L 的计算公式为：

$$L = \frac{1}{m}\sum_{i=1}^{m} d_i \tag{2-3}$$

网络的平均聚类系数为所有节点局部聚类系数的算术平均数，根据公式2-1，网络 G 的平均聚类系数 C 的计算公式为：

$$C = \frac{1}{n}\sum_{i=1}^{n} cn_i \tag{2-4}$$

等规模随机网络的平均路径长度 $\bar{L}$ 和平均聚类系数 $\bar{C}$ 可以用节点总数 n 和节点度的平均值 $\bar{k}$ 来近似计算，满足：

$$\begin{aligned} \bar{L} &= \frac{ln(n)}{ln(\bar{k})} \\ \bar{C} &= \frac{\bar{k}}{n} \end{aligned} \tag{2-5}$$

则小世界网络的判定条件为满足以下公式：

$$\begin{aligned} L &\approx \bar{L} \\ C &\gg \bar{C} \end{aligned} \tag{2-6}$$

2.3 图分析算法

由于在线社交网络及新闻传播过程可以使用有向图的形式进行表示，因此在本文中使用了图分析算法对其结构特性进行分析。本小节将分别介绍本文使用的两种图分析算法：SimRank 算法和 PageRank 算法。

2.3.1 SimRank 算法

SimRank 算法由 Glen 等人[79] 与 2002 年提出，是一种用来计算有向图中节点之间拓扑结构相似度的算法。相比其他计算相似度的方法，如直接比较节点的入度和出度，SimRank 算法只专注于有向图的拓扑结构，不依赖于节点的属性，使得 SimRank 算法能更好地衡量节点在结构方面的相似度。同时 SimRank 算法是递归定义的，这使得它可以捕捉到有向图更深层次的结构信息。

SimRank 算法的基本思想如下：对于有向图 G 中的任意两个节点 a,b，如果指向 a 的节点和指向 b 的节点相似，那么 a 和 b 也是相似的。SimRank 算法的递归起点是节点与它本身的相似度为 1。若设节点 a 的入边邻居节点集合为 $In(a)$，其中第 i 个节点为 $In_i(a)$，则节点 a,b 的相似度 $S(a,b)$ 递归地定义为：

$$S(a,b)=\begin{cases}1, & a=b\\ \frac{c}{|In(a)||In(b)|}\sum_i^{|In(a)|}\sum_j^{|In(b)|}S(In_i(a),In_j(b)), & a\neq b\wedge In(a),In(b)\neq\emptyset\\ 0, & otherwise\end{cases} \quad (2\text{-}7)$$

其中 $c\in[0,1]$ 是一个阻尼系数。由于 SimRank 算法是递归定义的，因此算法的复杂度较高，时间复杂度为 $O(|E|^2)$，空间复杂度为 $O(|V|^2)$，当图中节点数目很多时会耗费大量计算资源。因此，一些研究者提出了 SimRank 算法的近似算法。如 Fogaras 等人[80] 提出了一种基于蒙特卡罗模拟的 SimRank 随机算法，将 $S(a,b)$ 表示为分别从 a,b 出发进行随机游走相遇耗费的总时间的期望函数。这种方法降低了复杂度，但同时也降低了精度。

2.3.2 PageRank 算法

PageRank 算法由 Larry[81] 等人在 1996 年提出，用于衡量互联网网站页面的重要性。PageRank 算法受到了学术论文引文网络的启发，在学术论文中，衡量一篇论文质量的重要指标是它的被引用量。通常情况下引用量越高的论文质量越高。借鉴这种思想，PageRank 算法使用超链接的数目来衡量网页的重要性，同时还考虑了相邻网页重要性的影响。由于互联网页面组成的超链接网络可以被抽象为一个有向图，用节点代表网页，有向边代表超链接，因此 PageRank 算法可以被扩展应用于计算有向图中的节点在网络结构上的重要性。

有向图上 PageRank 算法的核心思想包括两点：

1. 对于一个节点 A，指向 A 的节点越多，则节点 A 的重要性越高，PageRank 值越高。
2. 对于一个节点 A，指向 A 的节点 PageRank 值越高，则节点 A 的重要性越高，PageRank 值越高。

由于指向 A 的节点的 PageRank 值也受到 A 的 PageRank 值影响，因此该

算法是循环定义的，无法直接计算。Larry 等人采用随机游走的方法，将节点 PageRank 值视为随机游走过程中节点被访问的概率，节点被访问的概率越高，则其 PageRank 值也越高。在给各节点赋予一个初始 PageRank 值的前提下，进行多次游走，到达每个节点后都以相等的概率跳转到它的出边邻居节点上。每次游走都使得各节点 PageRank 值不断更新，经过多次迭代最终得到收敛结果。PageRank 算法的具体过程如下：

对于有向图 $G = (V, E)$，其中 V 为节点集合，E 为边集合。设节点 $v_i \in V$ 的入边邻居节点为 $In(v_i)$，出边邻居节点为 $Out(v_i)$。首先，为 V 中每个节点赋予一个初始 PageRank 值 pr。对于 V 中的每个节点 v_i，若想通过随机游走到达 v_i，必须要从 v_i 的入边邻居节点出发。因此访问 v_i 的概率 $PR(v_i)$ 就为从它的入边邻居节点集合 $In(v_i)$ 中的各个节点访问 v_i 的概率之和。而对于 v_i 的一个入边邻居节点 v_j，由于是以等概率随机游走到其出边邻居节点集合 $Out(v_j)$ 中的任意一个节点，因而 v_j 下一步访问 v_i 的概率等于访问 v_j 的概率 $PR(v_j)$ 除以 v_j 的出边邻居节点个数 $|Out(v_j)|$。综上所述，在一次迭代后，节点 v_i 的新 PageRank 值 $PR'(v_i)$ 采用如下公式进行更新：

$$PR'(v_i) = \sum_{v_j}^{In(v_i)} \frac{PR(v_j)}{|Out(v_j)|} \tag{2-8}$$

当有向图 G 是一个强连通图时，PageRank 算法最终可以收敛。但当 G 中存在没有入边或出边的“孤立节点”时，因为随机游走会被困在“孤立节点”中，经过多次迭代会使得“孤立节点”的 PageRank 值逐渐变大，其他正常节点的 PageRank 值都逐渐趋近于 0。为了解决这种问题，PageRank 算法引入了一个小于 1 的常数 d，被称为阻尼系数，用于表示随机游走有目的地重定向到其他节点的概率，从而使得随机游走可以跳出孤立节点。修改后的 PageRank 算法，每次迭代的更新公式为：

$$PR'(v_i) = \frac{1-d}{|V|} + d \sum_{v_j}^{In(v_i)} \frac{PR(v_j)}{|Out(v_j)|} \tag{2-9}$$

通常情况下，d 取 0.85[81]。

2.4 神经网络模型

神经网络模型具有强大的学习能力，擅长学习数据中复杂的非线性映射关系。在本文的研究中使用了门控制循环单元搭建 WillCas 模型，并使用注意力机制提升模型性能。本小节将分别对这两种神经网络模型进行详细介绍。

2.4.1 门控制循环单元

门控制循环单元（GRU）是 Cho 等人[82] 于 2014 年提出的一种神经网络，是循环神经网络（RNN）的一种改进，旨在解决学习过程中的长期记忆和梯度消失等问题。相比 LSTM，GRU 的结构更加简单，这使得它拥有更高的训练效率，更擅长处理大规模数据集，同时具有与 LSTM 相似的实验效果。

与 RNN 相同，GRU 同样以当前节点 x^t 和上一个节点传递的隐藏状态 h^{t-1} 作为输入，输出当前节点的隐藏状态 h^t 和当前的预测结果 y^t。一次 GRU 迭代可以用以下公式来表示：

$$h^t, y^t = GRU(x^t, h^{t-1}) \tag{2-10}$$

相比 LSTM，GRU 对内部结构进行了精简，其内部结构如图2-1所示。GRU 只使用两个门来计算节点的隐藏状态，分别为更新门（upgrade gate）和重置门（reset gate）。

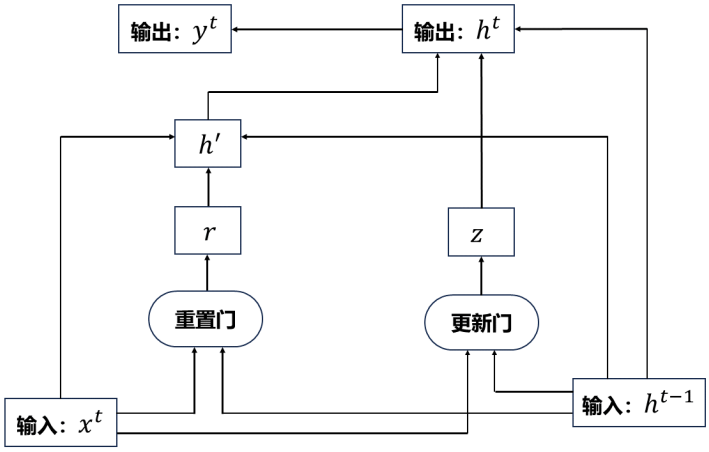


图 2-1　GRU 内部结构

GRU 的一次迭代具体过程如下：

首先，GRU 计算重置门的门控信号。重置门用于决定上一个节点传递的隐藏状态 h^{t-1} 中有多少信息要参与到计算当前节点的候选隐藏状态 h' 中。重置门的门控信号 r 的计算公式如下，其中 σ 为 sigmod 函数，下同：

$$r = \sigma(W^r \cdot \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}) \tag{2-11}$$

之后，GRU 根据重置门控信号 r 和当前节点输入 x^t，得到一个当前节点的候选隐藏状态 h'。节点的候选隐藏状态 h' 表示了当前节点的信息，将用于后续的

更新记忆步骤，其计算公式如下，其中 ⊙ 表示矩阵的元素积（哈达玛积），下同：

$$h' = tanh(W \cdot \begin{bmatrix} x^t \\ h^{t-1} \odot r \end{bmatrix}) \tag{2-12}$$

然后，GRU 计算更新门的门控信号。更新门用于决定在构建当前节点的隐藏状态 h^t 时，要遗忘多少上一节点的隐藏状态 h^{t-1}，同时又要记忆多少当前节点的候选隐藏状态 h'。更新门的门控信号 z 的计算公式为：

$$z = \sigma(W^z \cdot \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix}) \tag{2-13}$$

接下来，GRU 进入更新记忆阶段，通过选择性遗忘和记忆构造当前节点的隐藏状态 h^t，其计算公式为：

$$h^t = (1 - z) \odot h^{t-1} + z \odot h' \tag{2-14}$$

其中，$(1 - z) \odot h^{t-1}$ 表示对上一节点隐藏状态 h^{t-1} 的选择性遗忘，遗忘其中的一些不重要信息，$(1 - z)$ 表示遗忘的程度。$z \odot h'$ 表示对当前节点的候选隐藏状态 h' 的选择性记忆，记忆当前节点的一些重要信息，z 表示记忆的程度。

最后，GRU 输出当前节点的隐藏状态 h^t，同时还可以输出一个当前节点的预测输出 y^t，其计算公式为：

$$y^t = \sigma(W^o \cdot h^t) \tag{2-15}$$

在上述公式中，W, W^r, W^z, W^o 都为学习得到的参数矩阵。

2.4.2 注意力机制

注意力机制（Attention Mechanism）是深度学习中的一种优化方法，由 Bahdanau 等人[83] 在 2014 年提出。注意力机制模仿了人类视觉和大脑认知的机制，通过赋予输入数据不同的注意力权重，使得神经网络能更多关注输入数据中更加重要的部分，从而提升模型的性能和泛化能力。注意力机制被广泛用于机器翻译、语音识别等研究工作中。

注意力机制的基本思想如下：首先，计算输入序列中每个位置的注意力权重，注意力权重值反映了输入数据中各部分的重要程度。注意力权重的计算通常是基于输入数据本身以及模型定义的各种参数。在计算出注意力权重后，就可以对输入序列求注意力加权和，将每个位置的输入值与对应的注意力权重相乘，求和得到序列的加权表示，该加权表示能更好地突出输入序列中更重要的部分。因此，计算注意力权重是注意力机制的核心过程。

注意力机制的具体计算过程如下：对于输入序列 $H = \{h_1, h_2, ..., h_n\}$，定义一个查询向量 Q，用来从输入序列中挖掘各个位置的注意力权重。同时定义一个打分函数 $S(h, Q)$，用于计算序列中每个节点 h_i 与 Q 之间的相关性。以加性模型为例，其打分函数被定义为：

$$S(h, Q) = v^T tanh(Wh + UQ) \tag{2-16}$$

其中 v, W, U 均为学习得到的参数矩阵或向量。在使用打分函数计算出 h 与 Q 的相关性分数后，需要对其进行归一化。通常使用 softmax 函数进行归一化操作，softmax 函数可以通过归一化，将相关性分数转换为 $[0, 1]$ 的值，并使得所有元素的相关性分数之和为 1，从而生成一个概率分布。经过 softmax 函数归一化后，便得到了输入序列 H 的注意力权重向量 α。节点 h_i 的注意力权重 α_i 的计算公式为：

$$\alpha_i = softmax(S(h_i, Q)) = \frac{exp(S(h_i, Q))}{\sum_{j=1}^{n} exp(S(h_j, Q))} \tag{2-17}$$

最后，使用以下公式计算加权和，得到序列的表示 H'：

$$H' = \sum_{i=1}^{n} \alpha_i h_i \tag{2-18}$$

2.5 本章小结

本章介绍了本文工作中涉及的相关理论和技术。本章首先介绍了在线社交网络的定义和形式化表示方法，之后介绍了在线社交网络的两个特性：局部聚类系数和小世界网络。随后，本章介绍了本文使用到的图分析算法，包括 SimRank 算法和 PageRank 算法。最后，本章介绍了本文使用的神经网络模型：门控制循环单元和注意力机制。

第 3 章　基于传播意愿的在线社交网络新闻流行度预测模型

针对现有的新闻流行度预测方法中忽略用户主观传播意愿的问题，本章提出了一个基于传播意愿的在线社交网络新闻流行度预测模型 WillCas，用于预测在线社交网络上的新闻流行度，模型结构如图3-1所示。该模型综合考虑了用户传播意愿、时间特征、网络结构特征三个方面的影响因素，并使用一个端到端的注意力图神经网络框架进行实现。本章将从问题定义、用户传播意愿、时间和网络结构特征、模型构建、实验与评估等方面详细阐述该模型。

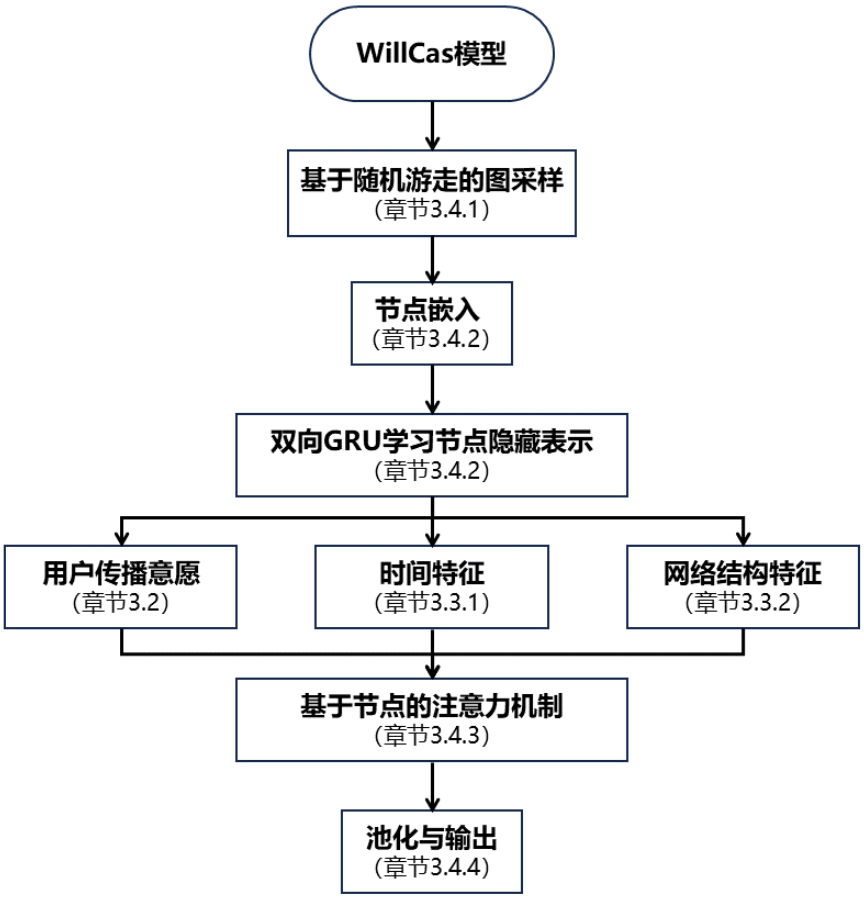


图 3-1　WillCas 模型整体结构

3.1 问题定义

本小节中将给出在线社交网络上新闻流行度预测问题的形式化定义。

本文使用有向图 $G = (V, E)$ 来表示在线社交网络上的一个用户网络。其中，V 为表示用户的节点集合，$v_i \in V$ 表示一个用户节点；$E \subset V \times V$ 为表示用户之间关注关系的有向边集合。$e_{ij} = (v_i, v_j) \in E$ 表示 v_i 对 v_j 建立了单向的关注关系，即 v_i 能接收到 v_j 发布的新闻。

对于在用户网络中传播的一条新闻 c，本文定义其初始用户节点为 v_c^0，定义已经转发新闻的用户为激活状态，尚未转发新闻的用户为非激活状态。非激活状态用户中，与激活状态用户相邻（存在关注关系），接受到了新闻但未进行转发的用户为待激活状态。待激活状态的用户通过转发操作变为激活状态，同时将关注了该用户的非激活状态用户变为待激活状态。用户状态的变化无法逆转。

在用户网络的基础上，本文定义新闻 c 在 t 时刻的传播网络为 $N_c^t = (V_c^t, E_c^t)$，其中 V_c^t 表示激活状态的用户节点集合，$v_c^i \in V_c^t$ 表示一个激活状态的用户节点，$V_c^t \subset V$；E_c^t 表示新闻传播路径的集合，$e_k = (v_c^i, v_c^j, t_k) \in E_c^T$ 表示新闻在 t_k 时刻由用户 v_c^i 传播给用户 v_c^j。

本文定义 WillCas 模型的任务目标为预测特定时间段内的增量流行度。根据上述定义，任务目标可以表示为在用户网络 G 上，对于一条新闻 c 在 t 时刻的传播网络 N_c^t，预测其在 Δt 时间段内的流行度增量：

$$\Delta P_c^{(t,\Delta t)} = |V_c^{(t+\Delta t)}| - |V_c^t| \tag{3-1}$$

为了实现任务目标，WillCas 模型基于以下几个假设：

1. **在线社交网络上的用户网络符合小世界网络特性。**根据 2.2 章节的介绍，在线社交网络的小世界网络特性使得用户网络中的大多数用户都可以接收到某一用户发布的新闻，这是新闻能够在用户网络中传播的重要条件。因此，本文提出的 WillCas 模型假设用户网络符合小世界网络特性。
2. **用户网络影响局部性假设。**即用户的行为只会受到自己和用户网络内其他用户的影响[84]。实际上，用户的行为可能会受到现实生活中其他因素的影响，如熟人推荐、传统媒体广告等等。由于无法获取相关数据并进行科学的量化表示，这些因素的影响难以衡量。因此，本文提出的 WillCas 模型只考虑用户自己以及用户网络内其他用户的影响。
3. **用户状态不可逆假设。**为了简化传播网络的结构，本文提出的 WillCas 模型假定用户一旦成为激活状态后就不会再回到非激活状态或待激活状态，即用户在转发一次新闻后就不会再转发来自其他用户的同一新闻。该假设

是基于客观实际设定的，在在线社交网络中，正常的用户很少会多次转发完全相同的内容[85]，因此，WillCas 模型中的用户只会从唯一一个前驱用户处转发新闻。此外，WillCas 模型还过滤了用户转发自己已转发新闻的行为。

3.2 用户传播意愿

用户传播意愿指的是用户在主观上愿意传播新闻的程度。在之前的工作中，研究者们大多只关注新闻传播过程中的网络结构特征和时间序列特征，忽视了作为新闻传播主体的用户的影响。

用户传播意愿主要受到两方面因素的影响：个人因素和他人因素。个人因素主要包括用户的习惯与偏好，例如用户转发新闻的频率，感兴趣的话题类型等等。其中，用户活跃程度是用户发布内容、参与互动的频率，是用户传播意愿的最直观体现。用户在在线社交网络上越活跃，证明用户在主观上越愿意发布或分享新闻内容。

他人因素则是在线社交网络中的其他用户的影响。其中，前驱用户造成的影响最为重要。前驱用户指的是将新闻传播给当前用户的上一用户，其自身的影响力反映了该用户的受关注程度。显然，用户更愿意从影响力大的用户处接收新闻，如在线社交网络上的官方媒体、知名博主等等。另一方面，Ghafari 等人[60]的研究表明，在线社交网络中的用户行为会受到更信任的用户的影响，用户更加愿意从信任的用户处传播新闻。因此，对前驱用户的信任程度也是影响用户传播意愿的重要因素。

因此，WillCas 模型从用户活跃程度、前驱用户影响力、基于用户画像的信任程度三个方面量化地计算用户的传播意愿。对于传播网络 N_c^t 中的某个待激活用户 v_c^i 和其关注的某个前驱激活用户 v_c^j，定义其传播意愿满足：

$$W_i = \omega_1 * A_i + \omega_2 * H_j + \omega_3 * T_{ij} \tag{3-2}$$

其中，W_i 表示用户传播意愿，A_i 表示用户活跃程度，H_j 表示前驱用户影响力，T_{ij} 表示基于用户画像的信任程度，ω_1、ω_2、ω_3 均为模型学习得到的权重参数，满足 $\omega_1 + \omega_2 + \omega_3 = 1$。

同时，WillCas 模型同样综合考虑了时间特征、网络结构特征的影响，如图3-2所示。本小节及 3.3 章节将分别详细介绍上述特征的计算方法。

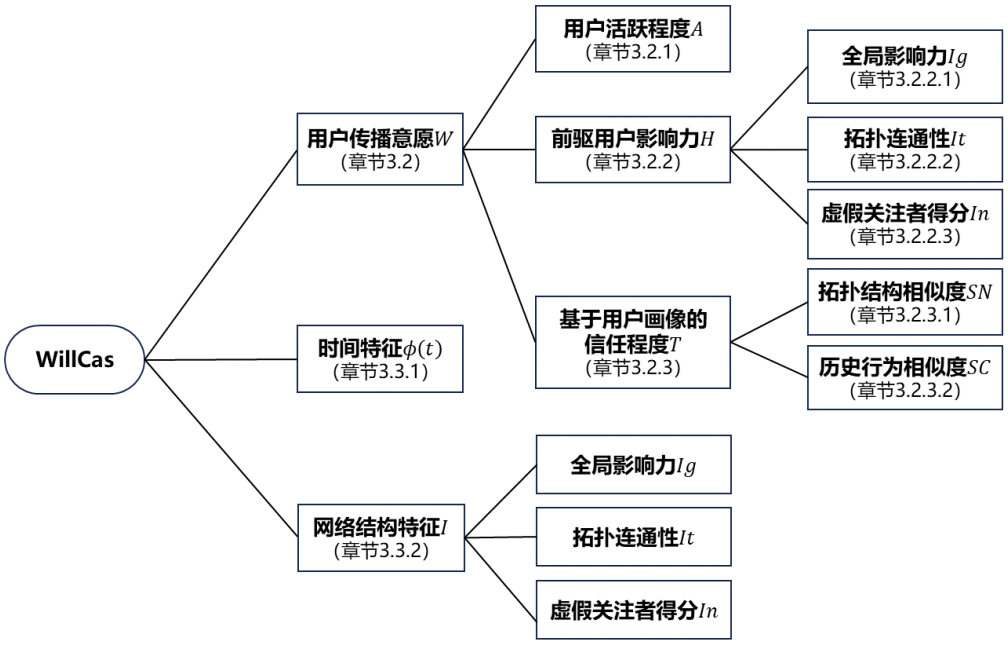


图 3-2　WillCas 模型考虑的特征

3.2.1　用户活跃程度

WillCas 模型使用用户发布和转发的历史新闻数量表示用户活跃程度。即对于用户 v_i 及其历史内容集合 C_{v_i}，用户活跃程度 A_i 可以表示为：

$$A_i = |C_{v_i}| \tag{3-3}$$

然而，由于在线社交网络中的长尾效应[86]，用户活跃程度呈现明显的不均衡性，只有少数用户非常活跃，如官方号、娱乐明星等，而大部分用户则不太活跃。如果直接使用历史新闻数量值作为活跃程度，可能导致用户活跃程度的差值过大，影响模型的结果。因此，WillCas 使用了最大-最小值归一化的方法，对历史新闻数量进行归一化后作为用户活跃程度的值。为了保证归一化后的活跃度值不为 0，在归一化要前先加上一个偏移量 ϵ（如 1×10^{-5})。修改后的公式为：

$$A_i = \frac{|C_{v_i}| + \epsilon - |C_v|_{min}}{|C_v|_{max} - |C_v|_{min}} \tag{3-4}$$

3.2.2　前驱用户影响力

传统的研究工作中大多采用用户粉丝数目、关注者数目等指标来简单衡量用户影响力，WillCas 模型则从全局影响力、拓扑连通性、虚假关注者得分三个方

面来计算用户影响力。对于在线社交网络中用户 v_i 的某个前驱用户 v_j，其影响力 H_j 的计算公式为：

$$H_j = Ig_j * It_j * In_j \tag{3-5}$$

其中 Ig 表示全局影响力，It 表示拓扑连通性，In 表示虚假关注者得分。接下来的三个小节将详细介绍各部分的计算方法。

3.2.2.1 全局影响力

用户的全局影响力指的是用户在整个用户网络中的重要程度。WillCas 模型使用网页排名算法（PageRank）来计算用户的影响力。根据 2.3.2 章节中的介绍，PageRank 算法可以用于计算有向图中节点的重要性。对于用户网络 $G = (V, E)$ 中的一个用户 v_i，定义其入边节点集合 $I(v_i) = \{v_j | (v_j, v_i) \in E \wedge v_j \in V\}$、出边邻居节点集合 $O(v_i) = \{v_j | (v_i, v_j) \in E \wedge v_j \in V\}$。则 v_i 的 PageRank 值计算公式为：

$$PageRank(v_i) = \frac{1-d}{|V|} + d \sum_{v_j \in I(v_i)} \frac{PageRank(v_j)}{|O(v_j)|} \tag{3-6}$$

其中 $d < 1$ 为预定义的阻尼系数，一般取 0.85[81]，$PageRank(v_i) \in [0, 1]$。则用户的全局影响力 Ig_i 为：

$$Ig_i = PageRank(v_i) \tag{3-7}$$

3.2.2.2 拓扑连通性

拓扑连通性指的是用户在用户网络中的局部连通性，体现了用户的相邻用户之间的连通情况，拓扑连通性越强的用户其邻居用户之间的联系越紧密。然而，强拓扑连通性在新闻传播中却通常对用户的影响力起负面作用[5]。因为当拓扑连通性较高时，用户的相邻用户之间可以不经过用户建立联系。换言之，用户在网络中是可有可无的，因为即使没有用户，信息也可以在相邻用户之间传播。如图3-3所示为一个用户网络，用户 u_3 和用户 u_5 拥有相同数量的邻居。用户 u_3 的拓扑连通性更强，它的邻居 u_1, u_2, u_4 之间的连通性更强；用户 u_5 的拓扑连通性较弱，它的邻居 u_4, u_6, u_7 之间的连通性也较弱。当去掉用户 u_3 时，信息仍然可以通过 $u_1 \rightarrow u_2 \rightarrow u_4 \rightarrow u_5$ 从网络左侧传播到网络右侧，u_8, u_9, u_{10} 等用户仍能收到信息。但如果去掉用户 u_5，信息将无法传播到网络右侧，$u_6, u_7, u_8, u_9, u_{10}$ 用户都无法收到信息。因此在该用户网络中，用户 u_5 的重要性比 u_3 更高。

WillCas 通过计算用户节点的局部聚类系数来衡量用户的拓扑连通性。局

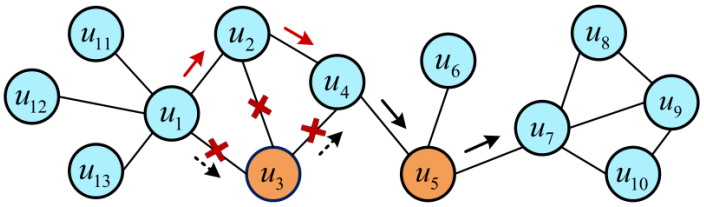


图 3-3　拓扑连通性负面影响示意图[5]

部聚类系数是用来衡量图中节点的邻居节点之间连接紧密程度的重要指标。一个节点的局部聚类系数越高，则节点的邻居节点之间连接越紧密，节点的拓扑连通性则越高。对于用户网络 $G = (V, E)$ 中的用户 v_i，定义其邻居节点集合 $N(v_i) = \{v_j | v_j \in V, (v_i, v_j) \in E \vee (v_j, v_i) \in E\}$，则用户节点 v_i 的局部聚类系数 cn_i 为：

$$cn_i = \frac{|\{(v_j, v_k) | j, k \in N(v_i), (v_j, v_k) \in E\}|}{|N(v_i)|(|N(v_i)| - 1)} \tag{3-8}$$

cn_i 的取值范围为 $[0, 1]$，为了表示拓扑连通性的负面影响，采用如下公式计算拓扑连通性 It_i：

$$It_i = e^{-cn_i} \tag{3-9}$$

3.2.2.3　虚假关注者得分

在在线社交网络中还存在一些用户，他们不是真实的活跃用户，而是出于一些目的而创建的虚假账号或机器人账号。这些账号活跃度很低，并且互相之间几乎没有建立关系[87]。如果有大量这样的虚假账号关注某个用户，那么他的影响力就可能被高估。因此，需要消除虚假粉丝带来的影响力。

WillCas 模型使用一个虚假因子 I_n 来表示用户的虚假关注者得分，得分越高表示用户的虚假关注者越少。根据这类虚假账号的特征，可以借助它们的全局影响力和拓扑连通性来计算 I_n。当用户的邻居节点影响力差且拓扑连通性弱时，表明这些用户更可能是虚假用户，则用户的虚假关注者得分越低。对于用户网络 $G = (V, E)$ 中的用户 v_i，定义其邻居节点集合 $N(v_i) = \{v_j | v_j \in V, (v_i, v_j) \in E \vee (v_j, v_i) \in E\}$，则通过以下公式计算 In_i：

$$
\begin{aligned}
num_i &= \sum_{j \in N(v_i)} \frac{|Ig_j|}{\sqrt{\sum_{k \in N(v_i)} |Ig_k^2|}} \\
con_i &= \sum_{j \in N(v_i)} \frac{|cn_j|}{\sqrt{\sum_{k \in N(v_i)} |cn_k^2|}} \\
In_i &= num_i * con_i
\end{aligned}
\tag{3-10}
$$

3.2.3 基于用户画像的信任程度

信任是一个复杂的社会心理概念，在线社交网络所提供的数据和信息中，并不包含可以直接体现用户信任程度的量化指标，需要设计合适的方法量化信任程度。社交同质性理论表明，拥有相似特征的用户之间更有可能存在联系[88]，相似的用户之间建立信任关系的可能性更高[63]。

因此，WillCas 模型使用用户之间的相似度来量化信任程度，通过建立用户画像的方式来体现用户特征，进而评估用户相似度。WillCas 模型从拓扑结构和历史行为两个方面建立用户画像，并分别计算相似度。对于某个待激活用户 v_i 及其某个前驱激活用户 v_j，定义 v_i 对 v_j 的信任程度满足：

$$
T_{ij} = \tau_1 * SN_{ij} + \tau_2 * SC_{ij} \tag{3-11}
$$

其中 T_{ij} 为用户 v_i 对用户 v_j 的信任程度，T_{ij} 的值越大表示信任程度越高。信任程度是单向的，T_{ij} 可能不等于 T_{ji}。SN_{ij} 为拓扑结构相似度，SC_{ij} 为历史行为相似度。τ_1、τ_2 为可调整的超参数，满足 $\tau_1 + \tau_2 = 1$。特别地，对于初始用户，由于不存在前驱激活用户，定义 $T_0 = 1$。接下来的两个小节将详细介绍各个相似度的计算方法。

3.2.3.1 拓扑结构相似度

用户网络的拓扑结构体现了用户的社交规模和互动强度，是用户传播能力的重要体现。通过拓扑结构的相似度可以有效比较用户在社交网络中的角色和作用。例如，意见领袖通常有较多的粉丝，但较少关注他人，其影响力往往更大，能够加速新闻的传播；而普通用户则相反，粉丝数目较少但较多地关注他人，其影响力也较弱。

WillCas 模型使用 SimRank 算法来计算用户之间的拓扑结构相似度。根据 2.3.1 章节的介绍，对于有向图 G 中的两个节点 a, b 以及其入边邻居节点集合 $I(a), I(b)$，定义 a 和 b 的相似度 $Sim(a, b)$ 满足：

$$Sim(a,b)=\begin{cases}1, & a=b\\ \frac{c}{|I(a)||I(b)|}\sum_{i}^{|I(a)|}\sum_{j}^{|I(b)|}Sim(I_i(a),I_j(b)), & a\neq b\wedge I(a),I(b)\neq\emptyset\\ 0, & otherwise\end{cases} \quad (3\text{-}12)$$

其中，$I_i(a), I_j(b)$ 分别为 $I(a), I(b)$ 的第 i, j 个元素，c 为常量衰减因子。

对于在线社交网络用户 v_i 和 v_j，根据公式(3-12)，计算其拓扑结构相似度 SN_{ij} 为：

$$SN_{ij}=Sim(v_i,v_j) \quad (3\text{-}13)$$

需要注意的是，由于 SimRank 算法是递归定义的，当用户网络的体量特别大时，SimRank 算法将耗费大量计算时间，还可能产生内存溢出现象。因此在实际应用过程中，通常使用基于蒙特卡罗模拟的 SimRank 算法来估算节点的 SimRank 值。基于蒙特卡罗模拟的 SimRank 算法使用随机游走的方法，通过计算从两个节点出发随机游走相遇的概率来估计相似度。这种方法虽然减少了计算量，但同时也降低了计算精度，其准确性依赖于随机游走的次数。随机游走的次数越多，准确性越高，但计算成本也会随之升高，因此需要合理调整随机游走的次数，平衡算法的准确性和效率。

3.2.3.2 历史行为相似度

用户的历史行为体现了用户使用在线社交网络的习惯，WillCas 模型使用 3.2.1 章节中提出的用户活跃程度来表示的历史行为。两个用户的活跃程度越相似，说明两个用户使用在线社交媒体的频率更加相似。对于在线社交网络用户 v_i 和 v_j，根据公式(3-4)，计算其用户历史行为相似度 SC_{ij} 为：

$$SC_{ij}=1-|A_i-A_j| \quad (3\text{-}14)$$

3.3 时间和网络结构特征

除主观因素外，客观条件也是影响在线社交网络上新闻传播的重要因素。本文提出的 WillCas 模型主要从时间特征和网络结构特征两个方面考虑客观因素的影响。本小节将分别详细介绍两种特征的计算方法。

3.3.1 时间特征

无论在哪种媒介中，新闻的传播的热度和范围都受到时间因素的影响。随着时间的推移，用户的新鲜感将下降，新闻的影响力也会不断减弱。在在线社交网

络平台中，新闻通常以时间倒序展示给用户，即用户最先看到的总是最近发布的新闻，之前发布的新闻会逐渐被移动到靠后的顺序，越来越难被用户发现。因此，在线社交网络中新闻的传播概率总体上会随着时间推移而呈现下降趋势[89]。

许多研究者使用概率分布函数来描述时间的影响。如 Wu 等人[33] 使用了指数函数进行回归预测，Litou 等人[90] 使用泊松分布来表示时间衰减效应。而 Zhao 等人[42] 通过分析大量推特平台真实数据，发现时间衰减效应更符合幂律分布，在 Gao 等人[35]、Mishara 等人[39] 的工作中也有着同样的发现。因此，WillCas 模型使用幂律分布来描述新闻传播的时间衰减效应，并且通过神经网络来学习其参数。对于当前时间 t，WillCas 定义时间衰减函数 $\phi(t)$ 为：

$$\phi(t) = at^{-b} \tag{3-15}$$

其中 a, b 为模型训练过程中学习得到的参数。

另外，由于受到人类生理活动昼夜节律的影响，用户在在线社交网络平台上的活跃度也在一天之内呈周期性变化[35]。例如，在凌晨睡眠时间用户的活跃度比夜晚黄金时间的用户活跃度要低很多。对于真实数据集来说，需要消除昼夜节律的影响，才能更好地表示时间衰减效应。因此，本文对实验数据进行了预处理，消除了昼夜节律的影响，详细方法在 3.5 章节中给出。此外，在更大范围内的周期变化（如星期、月、年）和一些特殊的时间变化（如节假日），由于较为复杂，不在本文的考虑范围之内。

3.3.2 网络结构特征

WillCas 模型使用用户自身在用户网络中的影响力作为网络结构特征的体现。参照 3.2.2 章节中介绍的方法，WillCas 模型同样通过计算全局影响力、拓扑连通性和虚假关注者得分来计算用户自身的影响力。根据公式3-5，用户影响力 I_i 的计算公式为：

$$I_i = Ig_i * It_i * In_i \tag{3-16}$$

其中 Ig_i、It_i、In_i 的计算方法已在 3.2.2 章节中给出，在此不再赘述。

3.4 模型构建

本文搭建了一个端到端的注意力图神经网络模型来实现 WillCas 模型。模型的结构如图3-4所示。模型以用户网络 $G = (V, E)$ 和新闻 c 在 t 时刻的传播网络 $N_c^t = (V_c^t, E_c^t)$ 为输入，输出在 Δt 时间段内的流行度增量 $\Delta P_c^{(t,\Delta t)}$。模型共分为四个

主要部分：

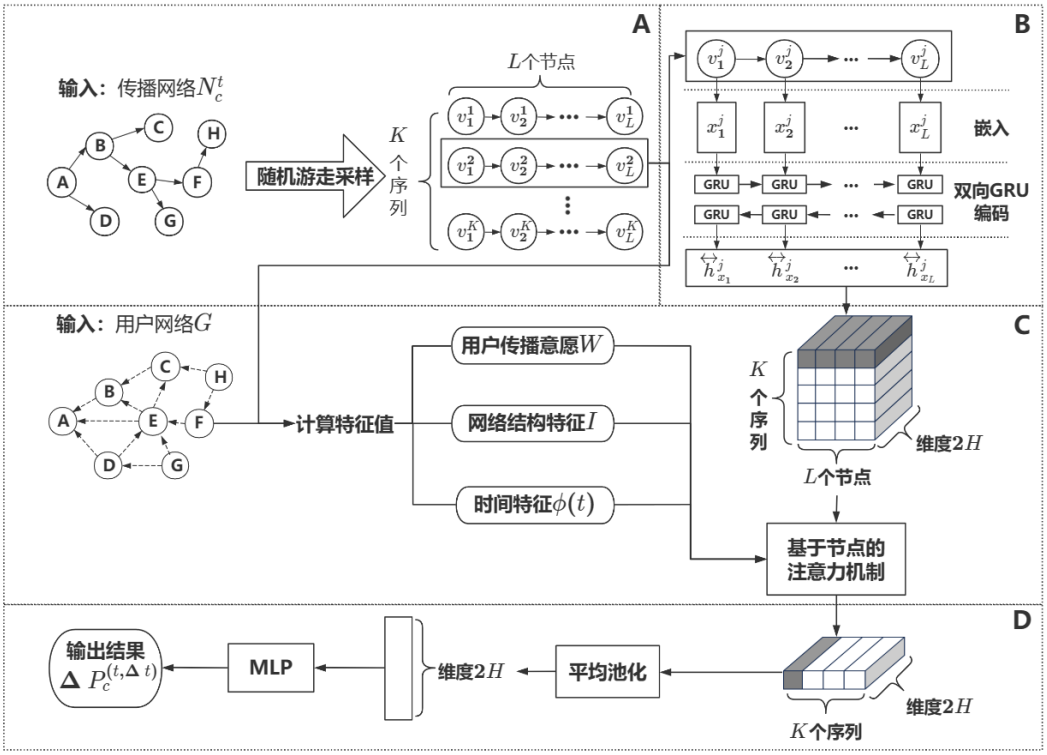


图 3-4　WillCas 模型结构图：A) 基于随机游走的传播网络采样 B) 节点嵌入与双向 GRU 编码 C) 基于节点的注意力机制 D) 池化与输出

1. **采样** (图3-4 *A*)。通过随机游走采样将传播网络采样为节点序列。
2. **嵌入与编码** (图3-4 *B*)。对节点进行嵌入，之后通过双向 GRU 学习节点的隐藏表示。
3. **基于节点的注意力机制** (图3-4 *C*)。引入用户传播意愿、时间特征和网络结构特征的影响，生成节点的注意力权重，计算加权和得到序列的表示。
4. **池化与输出** (图3-4 *D*)。对序列表示进行池化操作后得到传播网络的表示，输入 MLP，输出预测结果。

接下来本小节将详细介绍模型的各个主要部分。

3.4.1　采样

由于图结构的复杂性，WillCas 首先要对传播网络 $N_c^t = (V_c^t, E_c^t)$ 进行采样。为了不丢失传播网络中的结构信息，WillCas 从中采样一组节点序列。该方法借助了类比的思想，将传播网络类比为文档，用户节点类比为单词，将用户节点组成的序列类比为句子[7]，使用句子来代表文章的内容。

WillCas 使用随机游走算法进行序列采样，其采样过程的马尔可夫链如图3-5所示。采样过程从初始状态 S 开始，然后进入初始节点选择状态 J。在状态 J 中，算法从用户网络中随机选择一个节点作为初始节点。之后以 p_0 的概率进入

N 状态，开始采样一个序列；或者以 $1-p_0$ 的概率丢弃初始节点，进入终止状态 T，结束算法。在状态 N 中，算法从当前用户节点的出边邻居节点中随机选择一个节点跳转，并将该节点加入序列中。之后以 p_j 的概率结束当前序列的采样，进入 J 状态，选择新序列的初始节点；或者以 $1-p_j$ 的概率重复上述过程，继续采样新节点。

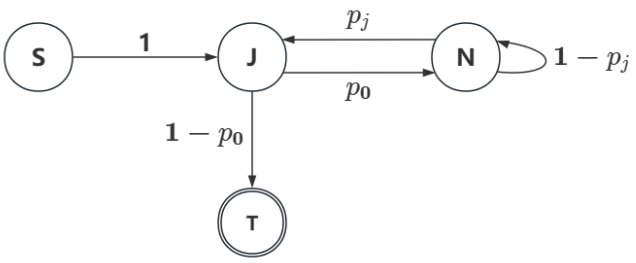


图 3-5 随机游走采样算法的马尔可夫链

在状态 J 中，传播网络 N_c^t 中的用户节点 v 被选择为初始节点的概率 $P(v)$ 为：

$$P(v)=\frac{d_c^t(v)+\epsilon}{\sum_{u\in V_c^t}(d_c^t(u)+\epsilon)} \tag{3-17}$$

其中 $d_c^t(v)$ 为用户节点 v 在用户网络 G 上的出度，ϵ 为平滑系数。

在状态 N 中，算法跳转到当前用户节点 v 的出边邻居节点 u 的概率 $P(u\in Out(v)|v)$ 为：

$$P(u\in Out(v)|v)=\frac{d_c^t(v)+\epsilon}{\sum_{u\in Out(v)}(d_c^t(u)+\epsilon)} \tag{3-18}$$

其中 $Out(v)$ 为 v 的出边邻居节点集合，满足 $Out(v)=\{u|u\in V_c^t\wedge(v,u,t_k)\in E_c^t\}$，$d_c^t(v)$ 为用户节点 v 在用户网络 G 上的出度，ϵ 为平滑系数。

在上述过程中，通过控制概率 p_j，可以控制每个采样序列的长度 L。通过控制概率 p_0，可以控制采样序列的个数 K。在本文中，WillCas 使用预定义的 K 和 L 进行采样。特别地，当算法处于状态 J 时，如果选择了孤立节点（度为 0 的节点）作为初始节点，则丢弃这个节点重新选择初始节点。当算法处于状态 N 时，如果选择跳转到了一个孤立节点，且序列长度未达到 L，则使用特殊节点“+”来填充剩余序列。

经过随机游走采样，WillCas 模型得到了 K 个长度为 L 的序列，用来表示传播网络 N_c^t。定义序列集合为 $S=\{s_1,...,s_K\}$，对 $\forall j\in[1,K]$，$s_j=[v_1^j,...,v_L^j]$。

3.4.2 嵌入与编码

为了学习采样序列的表示，首先要将序列中的节点嵌入为向量。对于采样序列集合 S 中的每个节点，将其表示为一个独热向量（one-hot 向量）q，其中 q 的维度为用户网络 $G=(V,E)$ 中的用户节点数量 +1（用于表示填充节点“+”），即 $q \in \mathbb{R}^{|V|+1}$。

由于用户网络中的用户数量较多，导致 q 的维度较高，占用大量空间，不便于后续的计算。因此，WillCas 模型使用一个嵌入矩阵 W_e 对 q 进行降维处理，得到一个低维密集向量 x，满足：

$$x = W_e q \tag{3-19}$$

其中 $W_e \in \mathbb{R}^{H\times(|V|+1)}$ 为训练过程中学习的嵌入矩阵，所有节点共享同样的 W_e。H 为可调节的维度参数，$x \in \mathbb{R}^H$。通过上述嵌入过程，第 j 个长度为 L 的采样序列 s_j 可以表示为 $s_j = [x_1^j, ..., x_L^j]$。

在对节点进行嵌入后，WillCas 模型使用双向 GRU 来学习节点的隐藏表示。当从左到右（前向）使用 GRU 学习序列表示时，GRU 可以不断使用序列中的节点更新隐藏状态。前向 GRU 体现了序列中后续节点对隐藏表示的不断丰富，在一定程度上模拟了新闻的传播过程。对于节点 x_n^j，前向 GRU 的一次隐藏状态的迭代可以表示为：

$$\overrightarrow{h}_{x_n}^{j} = GRU_f(x_n^j, \overrightarrow{h}_{x_n}^{j-1}) \tag{3-20}$$

其中，$\overrightarrow{h}_{x_n}^{j} \in \mathbb{R}^H$。

同时，WillCas 模型还额外使用一个从右至左（后向）的 GRU 学习序列的表示。后向 GRU 可以使得序列中的节点了解哪些节点会受到其影响，从而丰富其隐藏表示。对于节点 x_n^j，后向 GRU 的一次隐藏状态的迭代可以表示为：

$$\overleftarrow{h}_{x_n}^{j} = GRU_b(x_n^j, \overleftarrow{h}_{x_n}^{j+1}) \tag{3-21}$$

其中，$\overleftarrow{h}_{x_n}^{j} \in \mathbb{R}^H$。

最终，对双向 GRU 分别得到的隐藏状态进行拼接，得到节点 x_n^j 的隐藏表示 $\overleftrightarrow{h}_{x_n}^{j}$ 为：

$$\overleftrightarrow{h}_{x_n}^{j} = \overrightarrow{h}_{x_n}^{j} \oplus \overleftarrow{h}_{x_n}^{j} \tag{3-22}$$

其中，$\overleftrightarrow{h}_{x_n}^{j} \in \mathbb{R}^{2H}$。

经过上述嵌入和编码过程，第 j 个长度为 L 的采样序列 s_j 可以表示为 $s_j = [\overleftrightarrow{h}^j_{x_1}, ..., \overleftrightarrow{h}^j_{x_L}]$。

3.4.3 基于节点的注意力机制

根据 3.2 和 3.3 章节中的介绍，序列中的节点在传播意愿、时间特征、网络结构特征方面都存在差异，节点在序列中的重要性也因此有所不同。为了在学习过程中更好地表现节点重要性的区别，WillCas 模型引入了注意力机制。注意力机制（Attention Mechanism）是一种神经网络的优化方法。通过为输入数据赋予不同的注意力权重，模型能够更加关注于输入数据中与当前任务更相关的部分，从而提升模型的性能和泛化能力[83]。

在 WillCas 模型中，通过注意力机制，可以同时考虑节点在传播意愿、时间特征、网络结构特征三个方面的影响因素，从而得到更合理的节点表示，提升模型的预测性能。对于第 j 个长度为 L 的采样序列 $s_j = [\overleftrightarrow{h}^j_{x_1}, ..., \overleftrightarrow{h}^j_{x_L}]$，其中某个节点 $\overleftrightarrow{h}^j_{x_i}(i \in [1, L])$ 的传播意愿注意力权重 α^j_i、网络结构特征注意力权重 β^j_i、时间特征注意力权重 γ^j_i，满足：

$$\begin{aligned} \alpha^j_i &= W_i \\ \beta^j_i &= I_i \\ \gamma^j_i &= \phi(t_i) \end{aligned} \tag{3-23}$$

其中 W_i、I_i、$\phi(t_i)$ 分别为节点 $\overleftrightarrow{h}^j_{x_i}$ 对应的原用户节点 v_i 的用户传播意愿、网络结构特征、时间特征。t_i 为用户 v_i 成为激活状态的时间，即 $\exists(v_b, v_i, t_i) \in E^t_c \wedge v_b \in V^t_c$。$t_i$ 一定早于当前时间 t，且由于用户状态不可逆，因此只存在唯一的 t_i。

分别计算序列 s_j 中的所有 L 个节点的注意力权重，得到序列 s_j 的注意力权重向量 $\alpha_j = [\alpha^j_1, ..., \alpha^j_L]$、$\beta_j = [\beta^j_1, ..., \beta^j_L]$、$\gamma_j = [\gamma^j_1, ..., \gamma^j_L]$。之后，通过线性变换合并上述注意力权重向量，并使用 softmax 函数确保所有节点的注意力权重之和为 1。得到序列 s_j 的注意力权重向量 θ_j，满足：

$$\theta_j = softmax(W_a \times \begin{bmatrix} \alpha_j \\ \beta_j \\ \gamma_j \end{bmatrix}) \tag{3-24}$$

其中，$W_a \in \mathbb{R}^3$ 为学习得到的注意力权重合并向量，$\theta_j \in \mathbb{R}^L$。

最后，通过加权聚合得到第 j 个序列 s_j 的表示如下：

$$s_j = \sum_{i=1}^{L} \theta_i \overleftrightarrow{h}^j_{x_i} \tag{3-25}$$

其中，$s_j \in \mathbb{R}^{2H}$。

3.4.4 池化与输出

为了得到预测结果，需要生成传播网络的表示。WillCas 模型根据已有的序列表示集合 $S = \{s_1, ..., s_K\}$，使用平均池化的方法得到传播网络 N_c^t 的表示 $h(N_c^t)$，满足以下公式：

$$h(N_c^t) = \frac{1}{|S|} \sum_{j=1}^{K} s_j \tag{3-26}$$

其中 $h(N_c^t) \in \mathbb{R}^{2H}$。即将传播网络 N_c^t 使用一个 2H 维向量进行表示。

最后，将 $h(N_c^t)$ 输入一个 MLP，得到预测结果：

$$\Delta P_c^{(t,\Delta t)} = MLP(h(N_c^t)) \tag{3-27}$$

3.5 实验与评估

为了验证 WillCas 模型的预测效果，本文在真实数据集上进行了实验，通过与基准模型进行比对，评估模型预测的准确度。同时本文还进行了消融实验，用于验证各部分特征的有效性。

3.5.1 数据集

针对在线社交网络的应用场景，本文使用了两个新浪微博数据集来评估 WillCas 模型的效果。新浪微博是中国最受欢迎的在线社交网络平台之一。在新浪微博中，用户通过关注互相建立关系，粉丝会收到其关注者发布的微博，用户之间通过转发微博实现新闻的传播。

- **DeepHawkes 数据集**。DeepHawkes 数据集由 Cao 等人[1] 收集并公开发布。Cao 等人收集了新浪微博平台 2016 年 6 月 1 日生成的部分微博，并追踪在接下来 24 小时内的转发情况，为每条微博构建了包括转发路径、转发时间、参与用户 ID 等信息在内的传播级联。
- **SIL 数据集**。SIL 数据集由 Zhang 等人[91] 收集并公开发布。Zhang 等人在新浪微博平台中随机选择了 100 个种子用户，并爬取了他们的关注者和粉丝用户。对于每个用户，收集其最近发布的 1000 条微博，并追踪每条微博的转发情况，构建了级联数据。由于 SIL 数据集的规模过于庞大，本文对其进行了抽样，随机选择了部分用户，并只保留 24 小时内的转发数据。

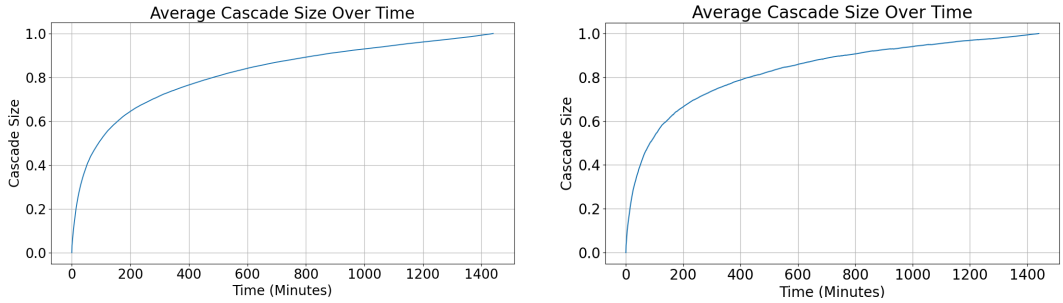
为了消除人类生理活动昼夜节律的影响，本文参考 Zhong 等人[5] 工作的思

想，只保留发布时间在 8:00 到 18:00 之间的微博。同时为了消除极端值的影响，本文还过滤了数据集中大小小于 10 或大于 1000 的级联数据。经过预处理后，根据级联数据构建了用户网络。数据集的规模和相关特征属性如表3-1所示：

表 3-1 数据集统计数据

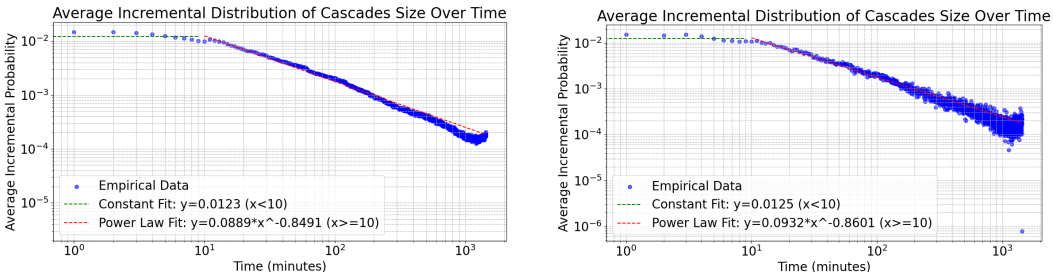
	节点数	边数	级联数	级联平均大小	级联平均深度
DeepHawkes	236867	277143	5094	70.1321	2.1699
SIL	446953	575500	10260	72.5751	2.2137

为了验证数据集的可用性，本文分别计算数据集中级联平均大小和平均增长率（增长量除以总增量），绘制出相关曲线如图3-6、3-7所示。可以发现，数据集中级联平均大小呈现幂函数增长趋势，平均增长率呈现开始是常数，之后逐渐衰减的趋势，与 Zhao 等人[42] 的工作相符，表明数据集符合在线社交网络的真实情况。



(a) DeepHawkes (b) SIL

图 3-6 标准化的级联平均大小



(a) DeepHawkes (b) SIL

图 3-7 级联平均增长率

3.5.2 对比实验

为了验证 WillCas 模型预测的准确性，本文进行了对比实验，与其他基准模型在相同数据集上的实验结果进行了比较，本小节将详细叙述对比实验的流程与

结果。

3.5.2.1 参数设置

对比实验中的 WillCas 模型参数设置如下：设定采样序列数目 $K = 200$，采样序列长度 $L = 10$，节点嵌入维度 $H = 50$。GRU 的隐藏层层数为 2，学习率为 0.001，嵌入矩阵 W_e 的学习率为 0.005。MLP 包含的全连接层数量为 2，维度分别为 32 和 16。随机游走采样的概率偏移量 $\epsilon = 0.01$，PageRank 算法的阻尼系数 $d = 0.85$，计算用户信任程度 T_{ij} 的超参数 $\tau_1 = 0.65, \tau_2 = 0.35$。训练过程中设置批量数据的大小 $batch\_size = 32$，设置 $dropout = 0.3$。对于数据集，按照 7:1.5:1.5 的比例划分训练集、验证集和测试集，连续 10 次迭代测试集上的损失函数值不再减少则终止训练过程。

根据图3-6中数据集级联平均大小的变化规律，在 3 小时之内级联的平均大小已经接近最大值的 70%。因此本文参照 Cao 等人[1] 工作中的设置，将观测时间分别设置为 1 小时、2 小时和 3 小时。

3.5.2.2 基准模型

根据 1.2.1 章节中的介绍，目前的级联预测方法中基于深度学习的方法要优于基于特征工程和过程模型的方法。因此，WillCas 模型的对比实验主要选取基于深度学习的方法进行对比。本文选取了五种基准模型，包括 DeepCas、DeepHawkes、CasCN、CoupledGNN 和 CasSeqGCN 模型。

- **DeepCas**[7]。DeepCas 是最早使用端到端深度学习的方法来预测级联大小的模型之一，同时引入了注意力机制来提升模型性能，但它主要依赖于级联中节点的结构和节点标签等信息，特征考虑较为简单。
- **DeepHawkes**[1]。DeepHawkes 是一种基于深度学习和过程模型相结合的方法，将霍克斯过程中用户影响力、自激励机制和时间衰减函数三个主要模块整合到一个深度学习模型中。
- **CasCN**[2]。CasCN 采用了一种先进的图采样方法，将原始级联图采样为多个子图，使用 GCN 来学习子图表示，并结合结构和时间特征，使用 LSTM 模型预测流行度。
- **CoupledGNN**[49]。CoupledGNN 创造性地使用了两个图神经网络，分别对用户节点的激活状态与影响力在用户网络中的扩散进行建模，学习二者的相互影响关系，得到用户的激活概率，并通过和池化计算流行度。
- **CasSeqGCN**[50]。CasSeqGCN 综合考虑了网络特征与时间特征，将级联划分为多个快照，并使用基于动态路由的节点表示聚合方法，利用图卷积网络 GCN 学习快照表示，并使用 LSTM 提取时间信息，预测流行度。

3.5.2.3 评价指标

本文使用平均绝对误差 MAE 和平均平方误差 MSE 来评估 WillCas 模型和各类基准模型的实验结果。

平均绝对误差 MAE（Mean Absolute Error）是用于衡量回归模型预测准确性的重要指标，代表了模型预测值与真实值之差绝对值的平均数。MAE 是预测误差的直观度量，其单位与原始数据相同。MAE 的值越小，代表模型的准确度更高。MAE 的计算公式为：

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3-28}$$

平均平方误差 MSE（Mean Square Error）是用于衡量回归模型预测准确性和稳定性的重要指标，代表了模型预测值与真实值之差平方的平均数。相比 MAE，MSE 由于计算了误差的平方，因此对异常值更敏感，更能体现模型预测的稳定性。MSE 的值越小，代表模型的准确度更高且误差波动更小。MSE 的计算公式为：

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3-29}$$

3.5.2.4 实验结果

表3-2、3-3为对比实验的实验结果。分析实验结果可知，相比于基准模型，WillCas 模型在预测准确性和稳定性上都有较好的表现。在预测准确度方面，WillCas 模型在所有实验中的 MAE 均优于所有基准模型。在预测稳定性方面，WillCas 模型除在 DeepHawkes 数据集观测时长为 2 小时的实验中和在 SIL 数据集观测时长为 3 小时的实验中，MSE 略高于 CasCN 模型外（优于其他基准模型），在其他实验中 MSE 均优于所有基准模型。

观察实验结果还可发现：对于同一数据集，当观测时间越长时，MAE 和 MSE 越低，表明 WillCas 模型的预测效果越好。这是由于更长的观测时间使得一些数据样本中级联大小更大，可供神经网络学习的信息更多，学习效果更好。同理，相比 DeepHawkes 数据集，SIL 数据集的规模更大，数据样本更多，在观测时间相同的前提下，WillCas 模型在 SIL 数据集上的实验结果优于在 DeepHawkes 数据集上的结果。

表 3-2　对比实验结果（MAE）

模型	DeepHawkes			SIL		
	1h	2h	3h	1h	2h	3h
DeepCas	1.822	1.633	1.624	1.579	1.542	1.411
DeepHawkes	1.505	1.302	1.288	1.422	1.220	1.139
CasCN	1.477	1.241	1.193	1.224	1.066	0.982
CoupledGNN	1.226	1.198	1.148	1.142	1.032	0.951
CasSeqGCN	1.346	1.213	1.175	1.268	1.135	1.097
WillCas	**1.172**	**1.052**	**0.959**	**1.126**	**1.019**	**0.943**

表 3-3　对比实验结果（MSE）

模型	DeepHawkes			SIL		
	1h	2h	3h	1h	2h	3h
DeepCas	2.936	2.667	2.592	2.736	2.447	2.413
DeepHawkes	2.646	2.366	2.323	2.562	2.388	2.279
CasCN	2.311	**2.149**	2.119	2.191	2.145	**2.029**
CoupledGNN	2.251	2.246	2.237	2.229	2.189	2.113
CasSeqGCN	2.383	2.194	2.156	2.230	2.186	2.065
WillCas	**2.224**	2.158	**2.112**	**2.147**	**2.134**	2.050

3.5.3　消融实验

为了进一步验证 WillCas 模型中考虑的用户传播意愿、时间、网络结构三方面特征的有效性，分析各部分特征对模型性能的影响，本文对 WillCas 模型进行了消融实验。分别去除 WillCas 模型中的用户传播意愿模块、时间特征模块和网络结构特征模块，得到 WillCas 模型的三个子模型：WillCas-NW、WillCas-NT、WillCas-NI。同时对于用户传播意愿模块，分别去除用户活跃程度、前驱用户影响力和基于用户画像的信任程度三部分特征，得到 WillCas-NWA、WillCas-NWH、WillCas-NWT 三个子模型。在其他参数设置和数据集设置不变的前提下，进行了消融实验，采用 MAE 为评价指标，比较子模型与原模型的实验结果。消融实验的结果如表3-4所示：

分析实验结果可知，各子模型的预测误差相比原模型明显增大。消融实验的结果表明 WillCas 模型的各部分特征具有一定的有效性和合理性。

表 3-4 消融实验结果（MAE）

模型	DeepHawkes			SIL		
	1h	2h	3h	1h	2h	3h
WillCas	**1.172**	**1.052**	**0.959**	**1.126**	**1.019**	**0.943**
WillCas-NW	1.298	1.146	1.014	1.256	1.280	1.114
WillCas-NWA	1.251	1.097	1.008	1.242	1.054	0.981
WillCas-NWH	1.196	1.065	0.978	1.160	1.055	0.969
WillCas-NWT	1.277	1.140	1.011	1.231	1.112	1.041
WillCas-NT	1.365	1.443	1.358	1.383	1.331	1.187
WillCas-NI	1.313	1.269	1.232	1.213	1.206	1.155

3.6 本章小结

本章提出了一个基于传播意愿的在线社交网络新闻流行度预测模型 WillCas，主要从用户传播意愿、时间特征和网络结构特征三个方面进行预测。在用户传播意愿方面，主要考虑了用户活跃程度、前驱用户影响力、基于用户画像的信任程度；在时间特征方面，使用幂律分布表示时间衰减效应；在网络结构特征方面，从全局影响力、拓扑连通性、虚假关注者三个角度进行计算。本章使用一个端到端的注意力图神经网络框架实现了 WillCas 模型，并在真实数据集上进行了对比实验与消融实验。对比实验结果表明：相比于其他基准模型，WillCas 模型可以较为准确地预测在线社交网络的新闻流行度，并且其预测结果的稳定性相对较好。同时，消融实验证明了 WillCas 模型各部分特征的有效性。

第 4 章　基于流行度的在线社交网络新闻传播过程分析方法

流行度虽然体现了新闻传播的规模，但无法反映新闻传播的具体过程，缺乏相关的细节信息，不能满足分析新闻传播过程的需要。为了在预测流行度的基础上更加全面地理解和分析在线社交网络上的新闻传播过程，本章提出了一种基于流行度的在线社交网络新闻传播过程分析方法 SimCas，该方法借助 WillCas 模型预测得到的流行度对新闻传播过程进行模拟，并采用可视化的方式进行分析，其结构如图4-1所示。

SimCas 方法首先对新闻传播过程进行模拟，通过一个带流行度约束的新闻传播过程模拟算法 PopSim，利用 WillCas 模型的流行度预测结果约束模拟过程，得到一次模拟的新闻传播网络及其生成概率。但是，PopSim 算法的模拟结果具有一定随机性，为了探索传播过程的一般性规律，SimCas 方法使用 PopSim 算法进行多次模拟，并设计了一个概率加权新闻传播网络生成算法 ProNet，将多次模拟结果聚合为一个概率加权新闻传播网络，便于后续分析。之后，SimCas 方法根据模拟结果，通过一个可视分析系统 SimVis 进行分析，SimVis 系统能够更直观地展示新闻传播的过程和相关上下文信息，辅助研究人员开展分析工作。本章通过真实数据集上的实验验证了模拟结果的准确性，并通过案例分析进一步验证了分析功能的有效性。

本章将分别从带流行度约束的新闻传播过程模拟算法、概率加权新闻传播网络生成算法、准确性验证、可视化分析等方面进行详细介绍 SimCas 方法。

4.1　带流行度约束的新闻传播过程模拟算法

为了分析在线社交网络上的新闻传播过程，SimCas 方法首先要对新闻传播过程进行模拟。本小节提出了一个带流行度约束的新闻传播过程模拟算法 PopSim。PopSim 算法首先计算当前时刻的用户传播概率，并使用流行度增量约束下一时间片的新增激活用户数量，从满足约束条件的情况中随机选择一种作为模拟结果，并根据用户传播概率计算该结果产生的条件概率，最终得到一种完整的传播网络模拟结果及其生成概率。本小节将首先给出传播过程模拟问题的形式化定义，之后介绍用户传播概率和带流行度约束的条件传播概率的计算方法，再给出算法的具体步骤及伪代码。

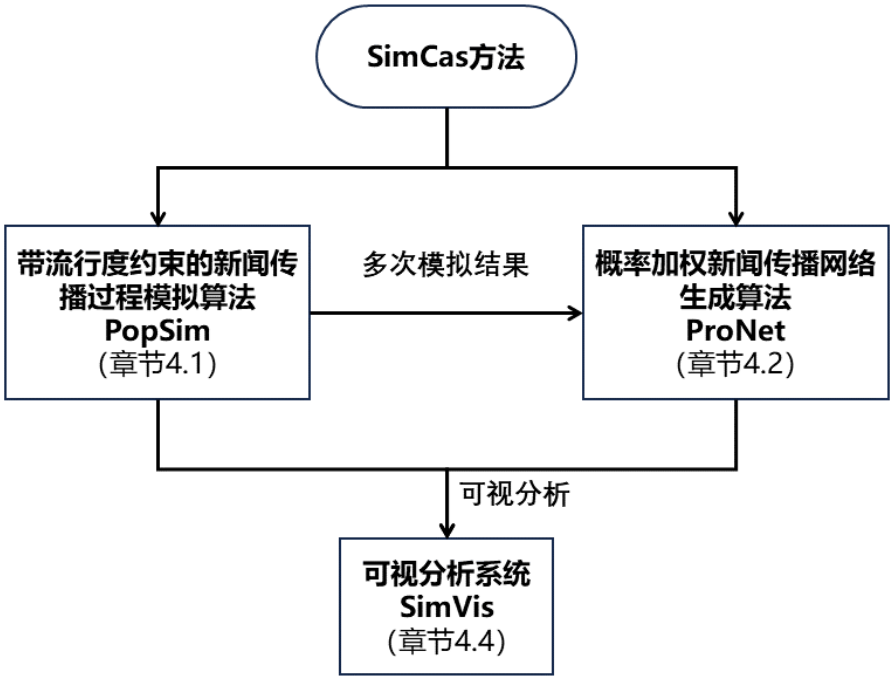


图 4-1　SimCas 方法整体结构

4.1.1　问题定义

本小节中将给出在线社交网络上新闻传播过程模拟问题的形式化定义。

PopSim 算法沿用 WillCas 模型中的相关定义，使用有向图 $G=(V,E)$ 来表示在线社交网络上的一个用户网络，使用 $N_c^t=(V_c^t,E_c^t)$ 表示新闻 c 在 t 时刻的传播网络。对于用户网络 G 中的某个用户节点 v_i，定义 $In(v_i)$ 为其入边邻居节点集合，满足 $In(v_i)=\{v_j|(v_j,v_i)\in E\wedge v_j\in V\}$，$In(v_i)$ 的大小等于 v_i 的入度。

同样地，PopSim 算法沿用 WillCas 模型中对用户状态的定义，将用户划分为激活状态、非激活状态、待激活状态三种类型。定义激活状态的用户节点集合为 X，非激活状态的用户节点集合为 X_u，待激活状态的用户节点集合为 X_w。X、X_u、X_t 三个集合满足如下关系：

$$\begin{aligned} &X,X_u,X_w\subset V\\ &X_w=\{v_i|(v_i,v_j)\in E\wedge v_i\in X_u,v_j\in X\}\subset X_u\\ &X\cap X_u=\emptyset,X\cup X_u=V \end{aligned} \tag{4-1}$$

为了简化传播过程，PopSim 算法在时间上将新闻传播过程视为离散过程，新闻以固定时间片 t_0 为步长进行传播。定义 t 时刻，预测得到的流行度增量为 $R_t=\Delta P_c^{(t,t_0)}$。

根据上述定义，新闻传播过程模拟的任务目标为：给定用户网络 G 和初始传

播网络 N_c^0，模拟到 t_e 时刻时，当前新闻 c 在 G 上的传播过程。

4.1.2 用户传播概率

用户传播概率指待激活状态用户从激活状态用户处转发新闻，转变为激活状态的概率。在 PopSim 算法中，每一时间片的新增用户数量已经由流行度进行约束，用户传播概率就决定了具体是哪些用户通过哪些路径被激活。PopSim 算法遵循 WillCas 模型的思想，从用户传播意愿、时间特征和网络结构特征三个方面计算用户传播概率。对于待激活状态的用户 v_i，定义 t 时刻被其关注的激活状态用户 v_j 激活的概率 $P(i,j)_t$ 为：

$$P(i,j)_t = a * W_i + b * \phi(t) + c * I_i \tag{4-2}$$

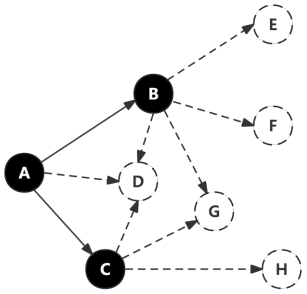
其中 W_i、$\phi(t)$、I_i 的计算方法遵循 3.2 和 3.3 章节中的定义，a、b、c 为可调整的参数，满足 $a+b+c=1$。特别地，为了保证计算得到的用户传播概率 $P(i,j)_t \in (0,1)$，需要对虚假关注者得分 In_i 进行归一化。

4.1.3 带流行度约束的条件传播概率

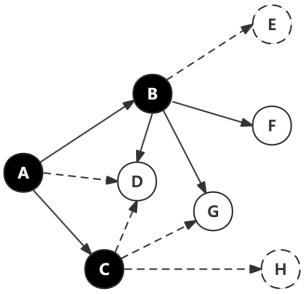
为了提升模拟结果的准确度，PopSim 算法使用预测得到的流行度来约束新闻传播模拟过程。对于传播过程中的每一时刻 t，在下一时间片 t_0 内新闻的流行度增量为 R_t，从而规定了下一时间片 t_0 内有且仅有 R_t 个待激活用户将被激活。这种约束限定了可能出现的模拟结果种类，改变了某种特定模拟结果出现的概率。因此，PopSim 算法引入了带流行度约束的条件传播概率来表示在下一时间片 t_0 内，某种特定模拟结果出现的概率。

带流行度约束的条件传播概率的定义为：对于一个待激活用户集合，且每个待激活用户对应至少一条传播路径。在已知流行度增量为 R_t，即有且只有 R_t 个待激活用户被激活，且每个用户只能被一条传播路径激活的条件下，选择了某组特定用户的某组特定传播路径的概率。

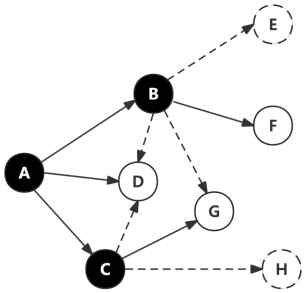
举例说明，如图4-2所示，图4-2 (a)是 t 时刻已有的传播网络，其中黑色节点代表激活用户，实线边代表已有的传播路径；虚线节点代表待激活用户，虚线边代表待激活用户的传播路径。此时待激活用户集合为 $\{D,E,F,G,H\}$，传播路径集合为 $\{AD,BD,CD,BE,BF,BG,CG,CH\}$。设在下一时间片 t_0 内的流行度增量为 3，则模拟结果被限制为只能从用户集合 $\{D,E,F,G,H\}$ 中选择 3 个用户，每个用户选择一条路径. 如图4-2 (b)、4-2 (c)所示为可能出现的两种模拟结果，虽然选择了同样的 3 个用户 D、F、G 被激活，但选择的传播路径不同，因此为两种不同的模拟结果。而4-2 (d)则是一种不可能出现的模拟结果，这种结果被



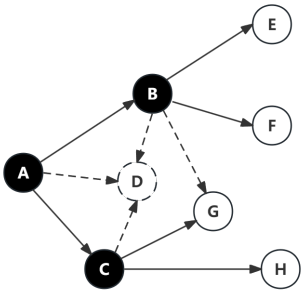
(a) t 时刻传播网络



(b) 可能的模拟结果 a



(c) 可能的模拟结果 b



(d) 不可能出现的模拟结果 c

图 4-2　带流行度约束的条件传播概率示例

排除在计算之外。因此，对于该示例，t 时刻带流行度约束的条件传播概率被定义为：从用户集合 $\{D, E, F, G, H\}$ 中选择 3 个用户，每个用户从传播路径集合 $\{AD, BD, CD, BE, BF, BG, CG, CH\}$ 选择一条路径的条件下，某种模拟结果（例如图4-2 (b)）的出现概率。

下面给出带流行度约束的条件传播概率的形式化定义。

给定用户网络 $G = (V, E)$。在 t 时刻，设激活用户集合为 X，待激活用户集合为 $X_w = \{v_1, v_2, ..., v_{|X_w|}\}$，可能的传播路径集合为 $N = \{(v_a, v_b)|v_a \in X, v_b \in X_w \wedge (v_b, v_a) \in E\}$。定义 $v_i \in X_w$ 在 N 中的传播路径子集合为 $N_v^i = \{(v_a, v_i)|(v_a, v_i) \in N\}$，在下一时间片 t_0 内的流行度增量为 R_t。

定义 t 时刻事件 A：在下一时间片 t_0，随机选择 X_w 集合中的 R_t 个用户传播，共有 L 种用户组合。对于每个用户，选择其任意一条传播路径, 共有 K 种路径组合。

定义 t 时刻事件 B_k：在下一时间片 t_0，选择了特定 R_t 个用户 $s_l = \{v_{l_1}, v_{l_2}, ..., v_{l_{R_t}}\}$ 传播，并对于 $\forall v_{l_n} \in s_l$，选择 $N_v^{l_n}$ 中的一条传播路径，形成了一组特定的传播路径 $N_s^k = \{n_{k_1}, n_{k_2}, ..., n_{k_{R_t}}\} \subset N$。

则 t 时刻带流行度约束的条件传播概率 $P(B_k|A)_t$ 定义为：

$$P(B_k|A)_t = \frac{P(AB_k)}{P(A)} \tag{4-3}$$

满足 $\sum_{k=1}^{K} P(B_k|A)_t = 1$。

下面给出 $P(B_k|A)_t$ 的计算方法：设路径 $n \in N$ 在 t 时刻的用户传播概率为 $P(n,t)$，计算方法遵循公式4-2。则 $P(AB_k)$ 的计算公式为：

$$P(AB_k) = \prod_{n}^{N_s^k} P(n,t) \times \prod_{n}^{N-N_s^k} (1 - P(n,t)) \tag{4-4}$$

对于事件 A，设随机从 X_w 集合中选择 R_t 个用户，共有 L 种组合，对应的用户组合集合为 $S = \{s_1, s_2, ..., s_L\}$。对于每个用户任意选择其任意一条传播路径，设最终形成的传播路径共有 K 种。则 L、K 满足以下公式：

$$\begin{aligned} L &= C_{|X_w|}^{R_t} \\ K &= \sum_{l=1}^{L} \prod_{i=1}^{R_t} |N_v^{l_i}| \end{aligned} \tag{4-5}$$

则 $P(A)$ 的计算公式为：

$$P(A) = \sum_{k=1}^{K} P(AB_k) \tag{4-6}$$

4.1.4 算法过程

PopSim 算法以用户网络 $G = (V, E)$、初始传播网络 $N_c^0 = (V_c^0, E_c^0)$ 、观测时长 t_e 为输入，输出一个传播网络 Res 和一个生成概率 P_e。其中 $Res = \{(v_i, v_j, t) | v_i, v_j \in V, t < t_e\}$ 表示模拟的传播网络，每个元素代表一条传播路径及对应的传播时间；P_e 代表 Res 模拟结果的生成概率。算法的大致流程如图4-3所示。

定义激活用户节点集合为 $X = V_c^0$，待激活用户节点集合 $X_w = \emptyset$，未激活用户节点集合 $X_u = V - X$；输出传播网络 $Res = E_c^0$，生成概率 $P_e = 1$；当前时刻为 t，用于存储传播路径的中间集合 $P_x = \emptyset$。

在当前时刻 t，首先构造待激活用户节点 X_w。PopSim 算法遍历激活用户节点集合 X 中的每一个用户节点 v_i 的入边邻居节点 $In(v_i)$，寻找其中尚未加入 X_w 的未激活节点 v_j，将其加入 X_w。并计算这条传播路径的用户传播概率 $P(v_i, v_j, t)$，将其加入集合 P_x。

之后，PopSim 算法获取 t 时刻的流行度增量 R_t，并从 X_w 中随机选择一组大小为 R_t 的用户节点集合 s_l，并对 s_l 中的每个用户，从 P_x 中选择对应的一条

传播路径组成一组传播路径集合 N_s^k。计算 N_s^k 的带流行度约束的条件传播概率 $P(B_k|A)_t$，并将概率累积到生成 P_e 上，即 $P_e* = P(B_k|A)_t$。

随后，PopSim 算法将 N_s^k 中的传播路径及当前时刻 t 添加到输出结果 Res 中，将 s_l 中的用户节点添加到 X 中，并从 X_u、X_w、P_x 中删除对应的元素，并将时刻增加时间片 t_0。同时，更新 P_x 中的概率值。

算法重复上述步骤，直到到达 t_e 时刻，或者未激活用户节点 X_u 为空集，则终止迭代，输出 Res 和 P_e。PopSim 算法的伪代码如算法1所示。

在 PopSim 算法的实际应用中，可以使用不同种类的流行度预测方法来计算每一时刻的 R_t 值。在本文中，PopSim 算法使用第 3 章中提出的 WillCas 模型计算 R_t。

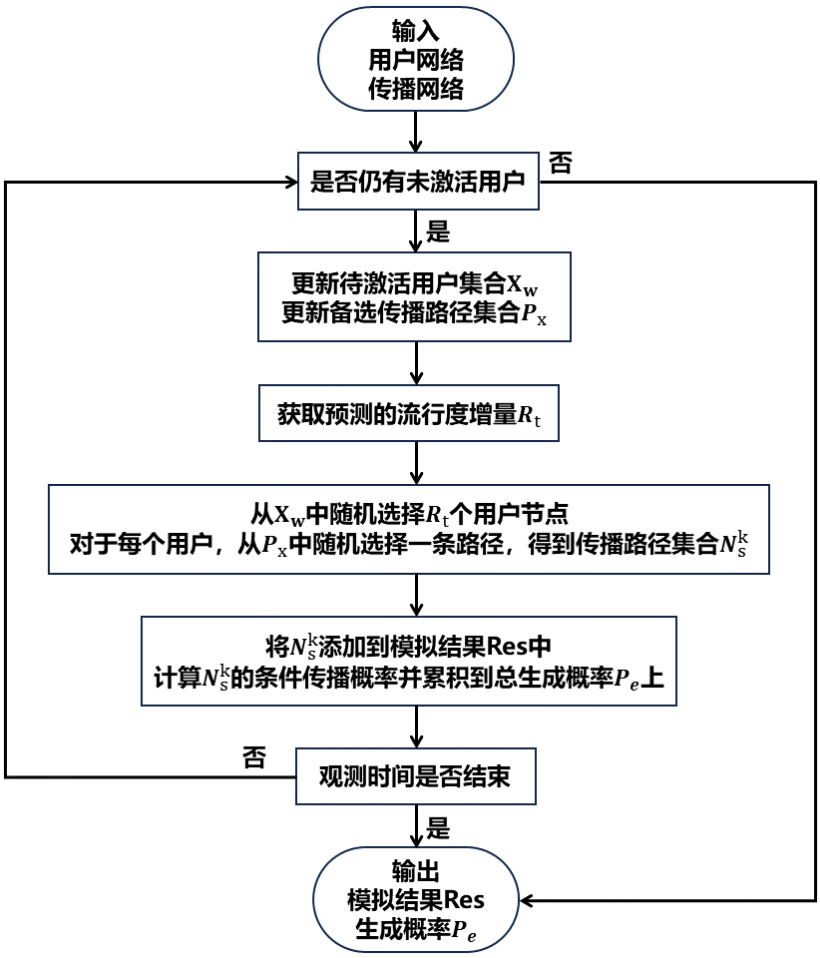


图 4-3　PopSim 算法流程示意图

设用户网络 G 中用户节点数量为 n，边数量为 m，节点的平均度数为 d，共进行了 t 次时间迭代。在每次迭代中，对于算法中构造待激活节点集合的操作，要遍历所有已激活节点的邻居节点，最坏情况下的时间复杂度为 $O(nd)$；对于更新 P_x 的操作，最坏情况下的时间复杂度为 $O(m)$；对于随机选择用户节点和传播

算法 1　带流行度约束的新闻传播过程模拟算法 PopSim

Require: $G=(V,E), N_c^0=(V_c^0,E_c^0), t_e$ ▷ 输入 G、N_c^0、t_e

Ensure: Res, P_e ▷ 输出传播网络 Res、生成概率 P_e

设激活用户节点集合 $X=V_c^0$，待激活用户节点集合 $X_w=\emptyset$，未激活用户节点集合 $X_u=V-X$；设输出传播网络 $Res=E_c^0$, 生成概率 $P_e=1$；设当前时刻为 t, 用于存储传播路径的中间集合 $P_x=\emptyset$

repeat

　　更新 P_x 中的概率值

　　for all v_i in X **do** ▷ 构造 X_w

　　　　for all v_j in $In(v_i)$ **and** v_j in X_u **and** v_j not in X_w **do**

　　　　　　X_w += v_j

　　　　　　P_x += $(v_i, v_j, P(v_i, v_j, t))$

　　　　end for

　　end for

　　$R_t = WillCas(t, t_0)$ ▷ WillCas 模型预测的流行度增量

　　从 X_w 中随机选择的一组用户节点 s_l

　　s_l 中的每个用户，从 P_x 中随机选择一条传播路径，组成传播路径集合 N_s^k

　　Pe *= $P(B_k|A)_t$ ▷ 累积生成概率

　　Res += (N_s^k, t) ▷ 将选中路径和时间添加到输出集合

　　X += s_l ▷ 将选中用户节点加入激活用户节点集合

　　X_u, X_w -= s_l ▷ 从未激活节点集合中移除选中用户节点

　　P_x -= N_s^k ▷ 从传播路径集合中移除选中路径

　　t += t_0

　　if $t=t_e$ **then**

　　　　break ▷ 当观测时间结束时结束算法

　　end if

until $X_u=\emptyset$ ▷ 当无未激活用户节点时结束算法

return Res, P_e

路径的操作，最坏情况下的时间复杂度为 $O(m)$；对于计算条件传播概率的操作，由于需要计算所有传播路径组合的生成概率，因此在最坏情况下的时间复杂度为 $O(2^m)$；对于算法中的其他操作，时间复杂度均为线性。综上所述，PopSim 算法在最坏情况下的时间复杂度为 $O(t(m + nd + 2^m))$。

通过上述分析可以发现，计算条件传播概率是 PopSim 算法的瓶颈。当传播路径组合数量较大时会导致组合数爆炸问题，计算时间会呈指数增长，甚至无法计算出最终结果。在实际应用中，结合 4.1.3 章节中条件传播概率的计算公式，可以采用基于蒙特卡罗模拟的方法，随机计算若干个组合的生成概率之和作为 $P(A)$ 的近似值，而不是计算所有组合的生成概率，从而降低算法复杂度。此外，对于拥有较多粉丝的用户节点，可以采用剪枝或聚类的方法，通过减少概率较小的传播路径数量来优化算法。

4.2 概率加权新闻传播网络生成算法

PopSim 算法虽然使用流行度约束了模拟过程，但得到的是满足约束条件的一种模拟结果，具有一定的随机性。为了探究新闻传播过程中的一般性规律，需要使用 PopSim 算法进行多次模拟，并将模拟结果组织为合理的形式，便于后续的分析工作。因此，SimCas 方法提出了一种概率加权新闻传播网络生成算法 ProNet，用于整合多次 PopSim 算法得到的模拟结果，计算传播网络中节点和路径出现的概率。本小节将首先介绍概率加权新闻传播网络的定义，之后详细介绍 ProNet 算法的具体过程。

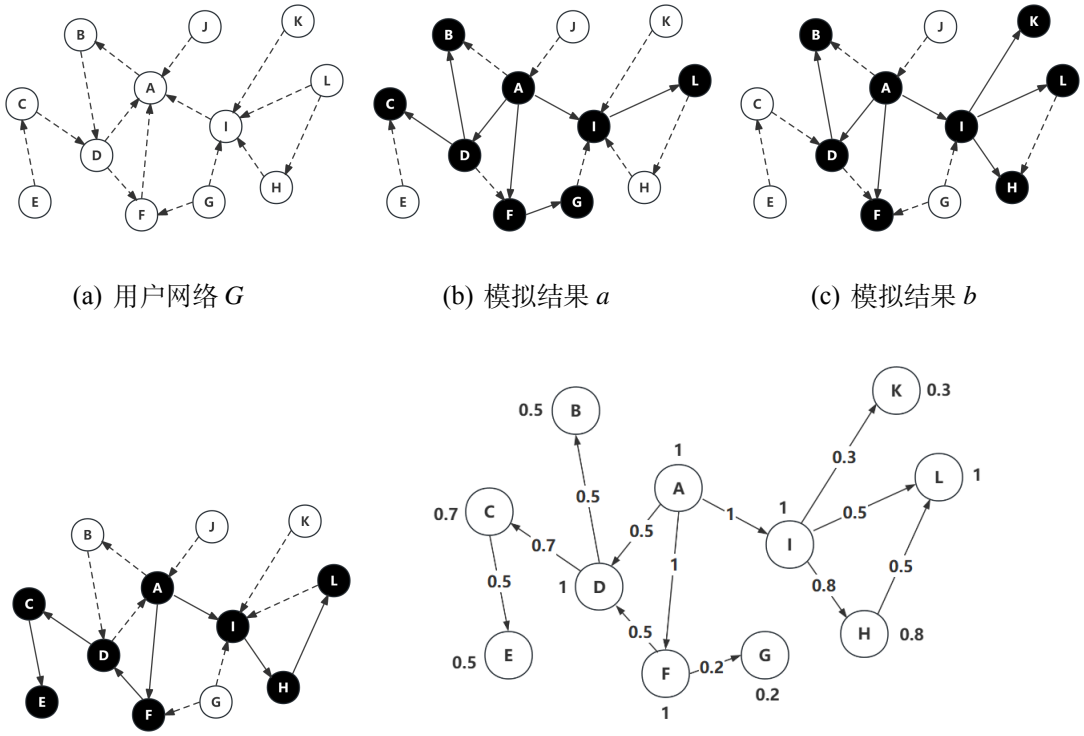
4.2.1 概率加权新闻传播网络

概率加权新闻传播网络建立在用户网络的基础上，包含所有模拟结果中出现的用户节点和传播路径。并且，该网络为所有用户节点和传播路径都赋予了一个权值，用来表示该节点或传播路径在所有模拟结果中出现的概率。概率加权新闻传播网络的形式化定义如下：

对于用户网络 $G = (V, E)$，设使用 PopSim 算法进行 n 次模拟后输出的传播网络集合为 $RES = \{Res_1, Res_2, ..., Res_n\}$，对应的生成概率集合为 $PE = \{P_{e_1}, P_{e_2}, ..., P_{e_n}\}$。其中 $Res_i = \{(v_i, v_j, t) | v_i, v_j \in V\}$。则定义概率加权新闻传播网络为 $G' = (V', E')$，其中 V' 为用户节点集合，满足 $V' = \{(v_i, w_v^i) | \exists j, k, t(v_i, v_j, t) \in Res_k \vee (v_j, v_i, t) \in Res_k\}$，即 V' 中的节点为 RES 中出现过的节点，w_v^i 为该节点出现的概率；E' 为传播路径集合，满足 $E' = \{(v_i, v_j, w_e^{ij}) | \exists k, t(v_i, v_j, t) \in Res_k\}$，即 E' 中的传播路径为 RES 中出现过的传播路径，w_e^{ij} 为该路径出现的概率。

举例说明，图4-4 (a)为给定的用户网络 G，其中虚线代表关注关系。图4-4 (b)、

4-4 (c)、4-4 (d)分别为 PopSim 算法的三次模拟结果 a、b、c，对应的生成概率分别为 0.2、0.3、0.5(本示例中为随意取值)，其中实线代表传播路径，黑色节点代表激活用户。则对应的概率加权新闻传播网络 G' 如图4-4 (e)所示。



(a) 用户网络 G　(b) 模拟结果 a　(c) 模拟结果 b

(d) 模拟结果 c　(e) 概率加权新闻传播网络 G'

图 4-4　概率加权新闻传播网络示例

4.2.2　算法过程

ProNet 算法以用户网络 G、传播网络模拟结果集合 RES、生成概率集合 PE 为输入，输出一个概率加权新闻传播网络 $G' = (V', E')$。算法的大致流程如图4-5所示。

设 $V' = \emptyset$，$E' = \emptyset$。遍历某个模拟结果 Res_i，对于 Res_i 中的一条传播路径 (v_{i_1}, v_{i_2}, t_i)，如果 v_{i_1}、v_{i_2} 不在 V 中，则将其加入 V'，并定义其概率权重为 Res_i 的生成概率 P_{e_i}。否则，如果 v_{i_1}、v_{i_2} 已经在 V 中，则将其概率权重加上 P_{e_i}。注意，在一次结果中，同一节点可能出现多次，应只累积一次概率权重。同理，若 (v_{i_1}, v_{i_2}) 不在 E' 中，则将其加入 E，并定义其概率权重为 Res_i 的生成概率 P_{e_i}。反之则将其概率权重加上 P_{e_i}。

对 RES 集合中所有模拟结果重复上述操作后，还需对概率权重进行归一化，保证所有概率权重取值范围在 [0,1] 之内，且能够很好地反映在所有模拟结果中的出现频率。因此，ProNet 算法将概率节点和边的概率权重除以所有模拟结果的

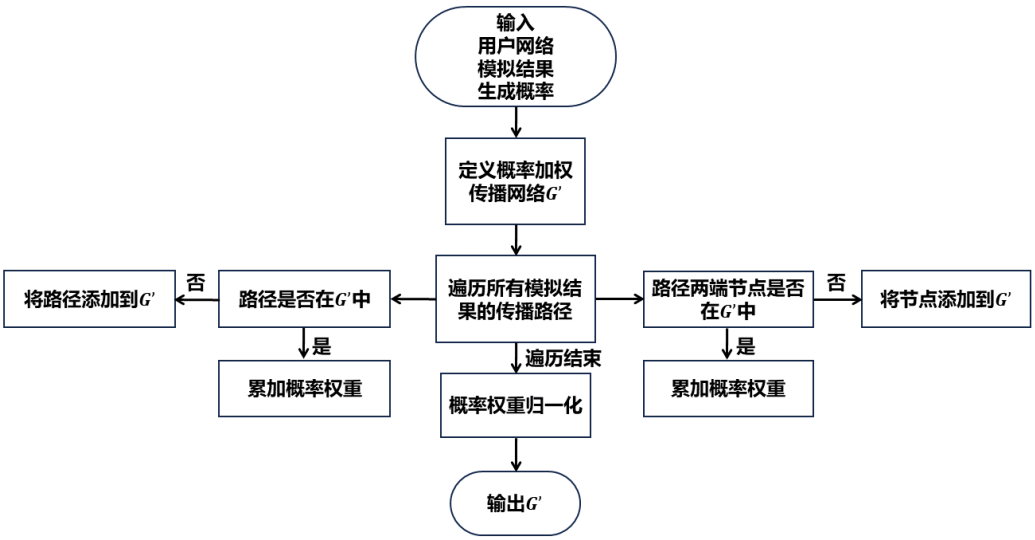


图 4-5　ProNet 算法流程示意图

生成概率之和，从而完成归一化操作。最后，输出概率加权传播网络 G'。ProNet 算法的伪代码如算法2所示。

设用户网络 G 中节点数量为 n，边数量为 m，模拟次数为 k。对于算法中的遍历操作，在最坏情况下，每次模拟中所有用户都被激活，一次模拟结果最多拥有 $n-1$ 条传播路径，则时间复杂度为 $O(nk)$；对于算法中的归一化操作，在最坏情况下，加权概率传播网络包括用户网络中的所有节点和边，则时间复杂度为 $O(nm)$。综上所述，ProNet 算法在最坏情况下的时间复杂度为 $O(nm+nk)$。

4.3　准确性验证

为了使用 SimCas 方法的模拟结果进行分析，需要先验证模拟方法的准确性，确保模拟结果符合真实情况。因此，本文通过一个准确性实验验证模拟结果是否符合真实数据集上的实际情况，并通过对比验证了流行度约束的有效性。本小节将从算法修改、实验设置、实验结果三方面进行详细介绍。

4.3.1　算法修改

准确性验证的核心目的是确保 PopSim 算法的设计是科学且准确的，即 PopSim 算法能够很好地模拟真实的新闻传播过程，其中用户传播概率的计算方法是准确的，使用流行度进行约束是有效的。因此需要将模拟结果与真实数据集上的数据进行比对。但由于 PopSim 算法在每个时间片内都是在流行度约束下随机选择一组用户节点及传播路径，因而得到的模拟结果不一定是生成概率最高的结果，不能直接用于比对。因此，本文在验证前先将 PopSim 算法修改为

算法 2　概率加权新闻传播网络生成算法 ProNet

Require: $G = (V, E), RES, PE$ ▷ 输入用户网络 G、模拟结果及生成概率 RES、PE

Ensure: $G' = (V', E')$ ▷ 输出一个概率加权新闻传播网络 $G' = (V', E')$

设用户节点集合 $V' = \emptyset$, 传播路径集合 $E' = \emptyset$
for all Res_i in RES **do**
    for all (v_{i_1}, v_{i_2}, t_i) in Res_i **do**
        if v_{i_1} not in V' **then** ▷ 节点不存在，添加节点
            V' += (v_{i_1}, P_{e_i})
        else if v_{i_1} not computed **then** ▷ 节点存在且没计算过，累加概率
            $w_v^{i_1}$ += P_{e_i}
        end if
        if v_{i_2} not in V' **then**
            V' += (v_{i_2}, P_{e_i})
        else if v_{i_2} not computed **then**
            $w_v^{i_2}$ += P_{e_i}
        end if
        if (v_{i_1}, v_{i_2}) not in E' **then** ▷ 边不存在，添加边
            E' += $(v_{i_1}, v_{i_2}, P_{e_i})$
        else ▷ 边存在，累加概率
            w_e^{ij} += P_{e_i}
        end if
    end for
end for
for all w in V', E' **do**
    w /= $sum(PE)$ ▷ 归一化
end for
return $G' = (V', E')$

PopSim-max 算法，用于获得生成概率最高的模拟结果。

PopSim-max 算法仅在 PopSim 算法的用户节点选择方法上进行了改动，应用了贪心算法的思想。对于每个时间片 t_0，PopSim-max 不再从 X_w 中随机选择一组用户节点及对应的传播路径，而是将所有可能的传播路径按其用户传播概率降序排序，选择前 R_t 个路径终点用户互不相同的路径进行传播（保证有 R_t 个用户被激活）。PopSim-max 算法的伪代码仅对 PopSim 伪代码中第 12 行、13 行改动为算法3所示：

算法 3　带流行度约束的真实新闻传播过程模拟算法 PopSim-max

1: sort(P_x)　　▷ 将 P_x 中所有的路径按用户传播概率降序排列

2: 从 P_x 中选择前 R_t 个终点用户互不相同的路径 N_s^k，对应的用户集合为 s_l，且满足 $|s_l| = R_t$

经过 PopSim-max 算法得到的传播网络 Res 是生成概率最高的传播网络，是最符合真实情况的模拟结果。

4.3.2　实验设置

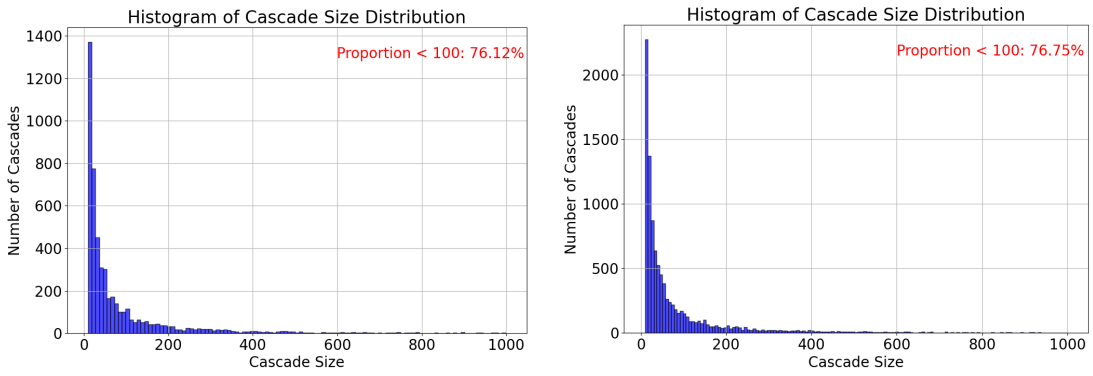
4.3.2.1　数据集

实验采用的数据集同样为 DeepHawkes 数据集和 SIL 数据集，数据预处理方式与 3.5.1 章节相同。由于算法复杂度的限制，较大规模的传播网络需要耗费大量计算资源，在实验前需要对数据样本进行筛选。图4-6为数据集中级联大小的分布情况，可以发现 75% 以上的级联大小都小于 100。因此，本文分别随机选取两个数据集中级联大小小于 100 的 1000 条数据样本进行实验，所选样本具有一定的代表性。

4.3.2.2　评价指标

实验采用 Jaccard 相似度作为评价指标。Jaccard 相似度是衡量两个集合之间相似度的指标，也可以用来比较两个有向图的相似度。本文同时计算边集合和节点集合的 Jaccard 相似度用于表示有向图的相似度，取值范围在 0 到 1 之间。相似度的值越大，表明算法的模拟效果越好。其计算公式为：

$$Jaccard(G1,G2) = \frac{1}{2}\left(\frac{|E1 \cap E2|}{|E1 \cup E2|} + \frac{|V1 \cap V2|}{|V1 \cup V2|}\right) \tag{4-7}$$



(a) DeepHawkes (b) SIL

图 4-6 级联大小分布情况

4.3.2.3 参数设置

图3-6中的数据显示，在传播开始后的 8 小时内，平均级联大小就已经超过了最大值的 80%。因此，本文设定 PopSim 算法的观测时长 t_e=8 小时，设定模拟时间片长度 t_0=300 秒。同时规定当原始数据样本级联中出现的节点全部被激活时提前停止模拟，防止因继续模拟产生的新激活节点导致相似度下降，影响评估效果。此外，对于每个数据样本重复进行 10 次模拟，计算相似度的平均值。

4.3.2.4 对比方法

除了计算模拟结果与真实结果的相似度之外，还需要验证 PopSim 算法中流行度约束的有效性，即使用流行度约束得到的模拟结果比真实结果的相似度更高。因此，实验选用以下两种方法进行对比，在对比方法中除流行度约束部分不同外，算法的其他部分均相同。

- **随机模拟算法 Random**。随机模拟算法在每个时间片内的流行度增量是完全随机的，任意待激活用户都有可能被激活，取决于对应路径的用户传播概率。Random 算法体现了去除流行度约束后的模拟效果。
- **基于统计流行度的模拟算法 Statistic**。基于统计流行度的模拟算法使用在数据集上的统计数据来作为流行度增量。对于每个时间片，流行度增量等于数据集平均级联大小的增量。Statistic 算法体现了使用其他类型流行度预测方法的模拟效果。

由于其他深度学习模型与 PopSim 算法使用的 WillCas 模型的流行度预测误差相对较小，在模拟中产生的差距较小，不易进行比较，因此不作为对比方法。

4.3.3 实验结果

实验结果如表4-1、4-2所示。分析实验结果可以发现，在模拟开始后的 1.5 小时内，PopSim-max 算法的模拟结果与真实结果的平均相似度已经超过 50%；随着时间的推移，PopSim-max 算法的准确性逐渐提升。到观测时间结束前，平均相似度已经超过 83%。同时，在每个观测时刻，PopSim-max 算法的模拟结果都要显著优于 Statistic 算法和 Random 算法。上述实验结果表明，对于当前规模的实验数据集，PopSim-max 算法具有较好的准确性，模拟的传播过程与真实情况较为相符，并且使用流行度约束模拟过程使得准确性得到了提高。

表 4-1　准确性实验结果（DeepHawkes）

时间	Random	Statistic	PopSim-max
0.25h	0.1626	0.1562	**0.2005**
0.5h	0.1671	0.2033	**0.2687**
1h	0.2932	0.3895	**0.4389**
1.5h	0.3651	0.4873	**0.5214**
2h	0.4228	0.5558	**0.5855**
4h	0.5608	0.5953	**0.6240**
6h	0.5953	0.7289	**0.7596**
8h	0.6240	0.7384	**0.7815**
12h	0.6387	0.8012	**0.8352**

表 4-2　准确性实验结果（SIL）

时间	Random	Statistic	PopSim-max
0.25h	0.0978	0.1421	**0.2132**
0.5h	0.1969	0.1956	**0.3056**
1h	0.2987	0.4007	**0.4325**
1.5h	0.3528	0.4875	**0.5069**
2h	0.4109	0.5630	**0.5824**
4h	0.5294	0.6833	**0.7030**
6h	0.5476	0.7230	**0.7321**
8h	0.5911	0.7655	**0.7872**
12h	0.6410	0.8051	**0.8396**

4.4 可视化分析

在使用 PopSim 算法模拟新闻传播过程，并使用 ProNet 算法将多次模拟结果聚合为一个概率加权新闻传播网络后，SimCas 方法借助一个可视分析系统 SimVis，使用可视分析的方式展示模拟结果和相关上下文信息，对传播过程进行分析。本小节将首先介绍 SimVis 系统的分析目标，之后介绍 SimVis 系统的各个组成部分，最后通过案例分析详细介绍分析过程。

4.4.1 分析目标

SimVis 系统的中心任务是分析在线社交网络新闻传播的过程，侧重于总览传播结构、追踪传播过程、分析传播差异性等方面。因此，本节对分析目标总结如下：

1. **总览新闻传播结构**。SimVis 系统应直观地展示概率加权新闻传播网络，便于研究人员了解任意时刻的新闻传播结构，了解当前时刻哪些用户通过哪些传播路径被激活，以及其相应的出现概率。新闻传播结构应易于理解，并尽可能避免出现视觉混乱。
2. **追踪新闻传播的动态过程**。SimVis 系统应能够动态地展示新闻传播网络随时间的变化情况，并提供相关上下文信息和交互方式。从而使得研究人员可以了解不同时间段内新闻的扩散过程，探索新闻传播的一般性规律，并分析传播路径被激活的原因。
3. **分析新闻传播中存在的差异性**。SimVis 系统应对用户特征等上下文进行直观展示，使研究人员可以对新闻传播在传播数量、路径生成概率、用户重要性等方面产生差异的原因进行分析。

4.4.2 系统介绍

SimVis 系统的用户界面如图4-7所示，本节将详细介绍 SimVis 系统各个模块的设计方式和使用方法。

4.4.2.1 加权网络视图

在加权网络视图中（图4-7（A）），SimVis 系统使用力导引图展示新闻传播网络的具体结构。首先，系统绘制了用户网络作为传播网络的基础。系统使用圆形节点表示用户，用带箭头的有向边表示用户的关注关系。之后，系统将 SimCas 方法模拟生成的概率加权传播网络显示到用户网络上，如图4-8 (a)所示。对于用户网络中的已激活用户节点，系统设置其透明度为正常，否则则设置为半透明状态。同样地，对于已激活的传播路径，系统使用实线表示新闻的传播方向；对于

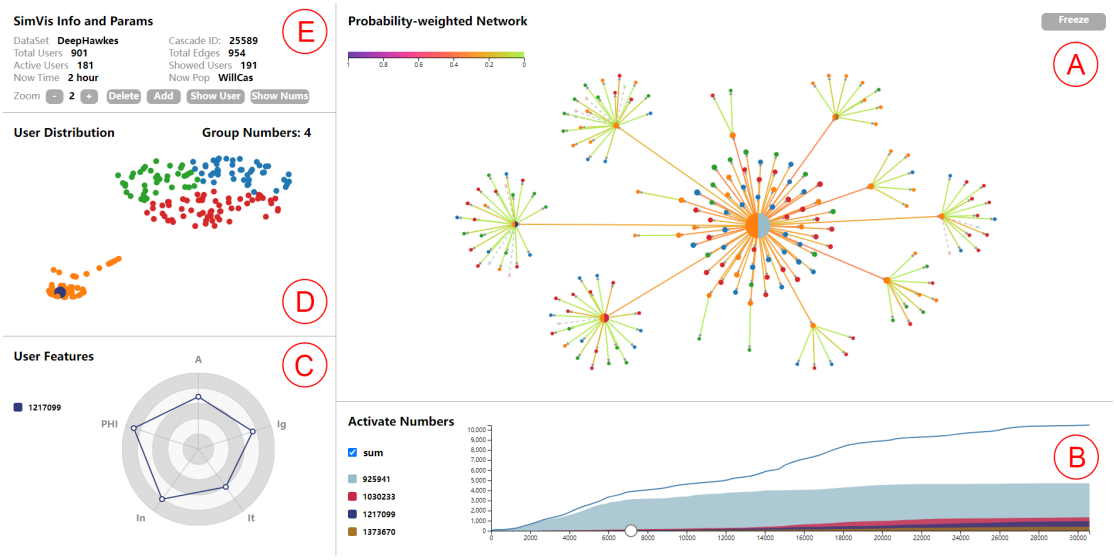


图 4-7　SimVis 系统的用户界面图：A) 加权网络视图展示了 SimCas 方法生成的概率加权新闻传播网络，并同时显示了用户网络；B) 时间轴视图用于展示用户传播的新闻数量随时间的变化关系，同时还用于控制当前系统时间 C) 用户特征视图可以显示用户或传播路径的相关特征；D) 用户分布视图将用户特征投影到二维平面上，并将具有相似特征的用户聚合到一起。E) 信息与参数视图列出了系统当前的各种参数和信息，并提供了交互面板。

未激活的传播路径，系统使用虚线表示用户之间的关注关系。

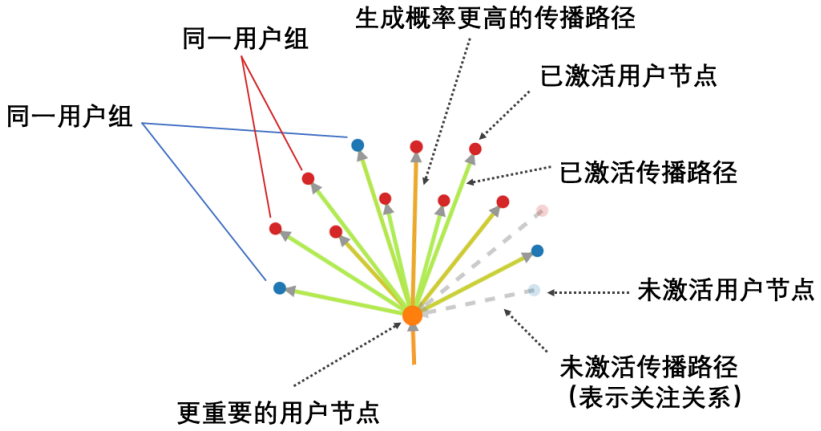
如图4-8 (a)所示，SimVis 系统对加权网络视图中的边和节点进行可视化编码。对于已激活的传播路径，系统设计了一个颜色比例尺来表示传播路径的生成概率，颜色越冷（偏向紫色）表明传播路径的生成概率越高，该路径在所有模拟结果中出现的次数越多，反之同理；对于未激活的传播路径则不编码任何颜色。对于用户节点，系统使用不同的颜色编码用户组，具有相同颜色的节点被划分为同一个用户组，与用户分布视图中的用户组颜色相对应。同时，对于已激活的用户节点，系统使用生成概率和该用户传播的新闻数量编码用户节点的大小，用于衡量用户节点在传播网络中的重要性。当用户的生成概率更高，且传播的新闻数量更多时，该用户在传播网络中更重要，其节点大小越大。

为了便于进行分析，系统允许研究人员对节点和传播路径进行标记，便于查看相关的上下文信息。如图4-8 (b)所示，当研究人员标记传播网络中的一个节点时，系统会为其分配一种标记颜色，并将该标记颜色与节点原有颜色共同显示在节点上。同时，在时间轴视图、用户特征视图和用户分布视图中都会使用相同的标记颜色指示该节点的相关信息。当研究人员标记一条传播路径时，路径的宽度会增加，进行突出显示。

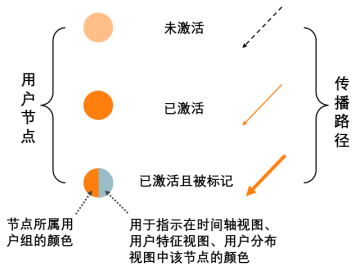
4.4.2.2　时间轴视图

在时间轴视图中（图4-7（B）），SimVis 系统使用一个重叠面积图来展示选定用户传播的新闻数量随时间的变化关系。如图4-8 (d)所示，当研究人员在加权网

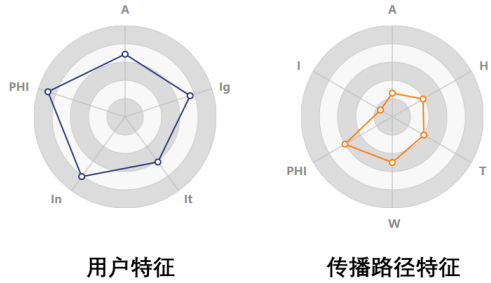
络视图中标记了某个用户节点后，系统便统计在所有模拟结果中，每个时刻该用户累计传播的新闻数量，并使用一个面积图将其绘制在坐标轴上。当研究人员同时标记了多个用户节点时，系统便将多个面积图重叠在一起。系统还允许研究人员选择是否显示所有节点传播新闻总数的曲线图，同时还可以通过拖动 x 轴上的圆形滑块，控制系统的当前时间。



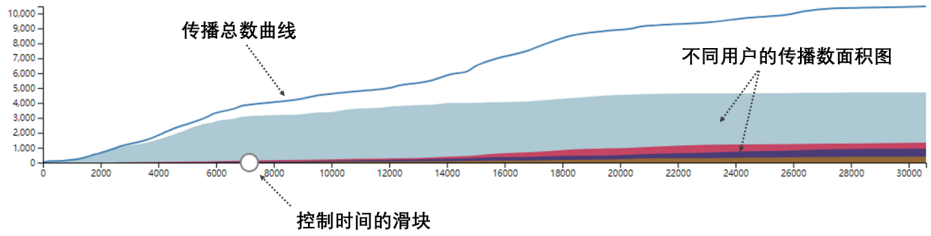
(a) 加权网络视图的可视化编码



(b) 用户节点的类型



(c) 用户特征视图的两种类型



(d) 时间轴视图

图 4-8 SimVis 系统的视图介绍

4.4.2.3 用户特征视图

在用户特征视图中（图4-7（C）），系统使用雷达图展示各类特征。如图4-8 (c)所示，当研究人员在加权网络视图中选中用户节点后，系统显示包括活跃程度 *A*、全局影响力 *Ig*、拓扑连通性 *It*、虚假关注者得分 *In* 和时间特征 *PHI* 在内的用户特征；当选中传播路径后，则显示该路径的用户活跃程度 *A*、前驱用户影响力 *H*、基于用户画像的信任程度 *T*、用户传播意愿 *W*、时间特征 *PHI*、网络结构特征 *I*。需要注意的是，由于特征数量不同，用户特征视图不能同时显示用户和传播路径的特征。

4.4.2.4 用户分布视图

在用户分布视图中（图4-7（D）），系统使用 t 邻域分布嵌入算法（t-SNE），将用户的特征投影到二维平面上，再通过 K 均值聚类算法（K-means）用户分成若干个用户组，并为每个用户组赋予一种颜色。当研究人员在加权网络视图中标记了某个用户节点后，用户分布视图也会使用相同的标记颜色突出显示该用户。在实际分析中，可以根据任务需要设定划分的群组数量。

4.4.2.5 信息与参数视图

信息与参数视图（图4-7（E））列出了 SimVis 系统的一些重要信息，包括使用的数据集名称、样本级联 ID、样本用户网络中的节点总数和边总数、当前被激活的用户数量、加权传播视图中显示的用户数量、当前系统时间和当前使用的流行度预测方法，并提供了一个交互面板。

4.4.3 案例分析

在 SimVis 系统的帮助下，研究人员得以通过可视分析的方式对在线社交网络上的新闻传播过程进行分析。本小节将通过案例分析详细介绍使用 SimVis 系统进行分析的具体过程，同时进一步验证 SimCas 方法分析功能的有效性。案例分析的三个任务对应 SimVis 系统的分析目标，分析过程使用了 DeepHawkes 数据集中的样本数据。

4.4.3.1 任务一：总览新闻传播结构

在分析新闻传播过程的任务中，研究人员首先通过 SimVis 系统观察新闻传播网络的总体结构，对新闻传播的整体情况进行把握，为后续分析寻找切入点。例如，图4-9为某数据样本传播开始 8 小时后生成的传播网络，用户 A 为初始用户，是新闻的发布者，用户 B、C、D 均为用户 A 的直接关注者，并且是各自子

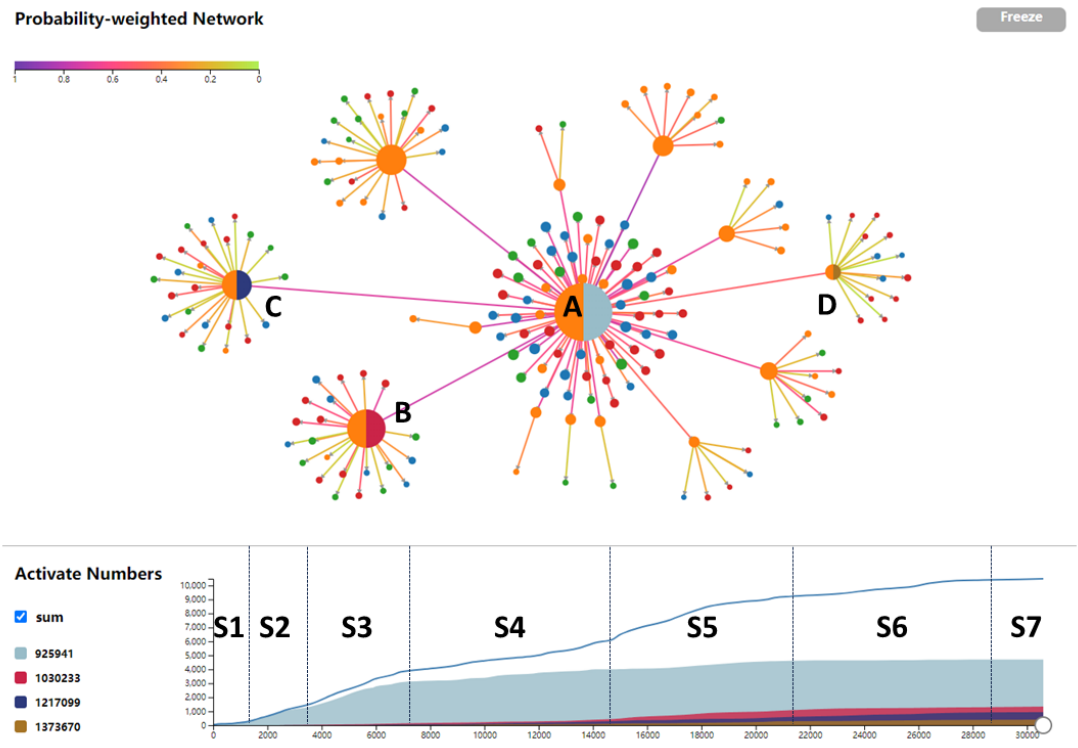


图 4-9 任务一：总览新闻传播结构

网络的中心用户。

通过观察时间轴视图可以发现，**新闻传播的过程呈现明显的阶段性**。新闻传播总量和不同用户的新闻传播数量在不同时间段呈现出不同的变化特点。例如，新闻传播总量在 S2、S3、S5 阶段的增长速率更快；用户 A 传播数量的增长主要集中于 S1-S3 阶段。

另一方面，通过观察加权网络视图可以发现，**新闻传播的过程中存在明显的差异性**，体现在以下几个方面：

1. **直接关注者和间接关注者之间的差异性。**例如，对于用户 A 的直接关注者，其传播路径的颜色均为粉红色或紫色，表明其生成概率大部分约在 0.7 以上；而对于用户 A 的间接关注者（第二跳关注者），其传播路径的颜色多为黄绿色或橙色，少部分为粉色，表明其生成概率相对较小，约在 0.1 到 0.7 之间。
2. **不同用户组之间的差异性。**例如，子网络的中心用户都属于被标记为橙色的用户组；并且在同一子网络内，属于红色和橙色用户组的传播路径生成概率要高于其他用户组。
3. **不同子网络之间的差异性。**例如，在用户 B 和 C 的子网络中，许多用户的传播路径的颜色为橙色或粉色，表明其生成概率较高，约在 0.4 到 0.7 之间；而在用户 D 的子网络中，所有用户的传播路径则均为黄绿色，其生成

概率均较低，约在 0.1 到 0.4 之间。并且，对于子网络规模相似的用户 B 和 C，B 的重要程度却明显高于 C；而用户 D 的重要程度远小于 B 和 C。

通过上述分析得到的初步结论，揭示研究人员可以进一步追踪新闻的动态传播过程，分析差异性的产生原因。在接下来的任务二和任务三中，将分别从这两个方向进一步进行分析。

4.4.3.2 任务二：追踪新闻传播的动态过程

通过任务一中的分析可以发现，新闻传播过程呈现阶段性。本小节将通过追踪新闻传播的动态过程，对新闻传播过程各个阶段的特点进行分析。在分析过程中，本小节重点关注任务一中标注的四个典型用户：

1. **用户 A（925941，标记为浅蓝色）**：传播网络的初始用户。
2. **用户 B（1030233，标记为粉红色）**：最大规模子网络的中心用户，重要程度仅次于 A，子网络内用户传播路径的生成概率普遍较高。
3. **用户 C（1217099，标记为藏蓝色）**：规模与 B 相似的子网络的中心用户，但重要程度明显低于 B。
4. **用户 D（1373670，标记为棕色）**：规模较小子网络的中心用户，子网络内用户传播路径的生成概率普遍偏低。

其中用户 B、C、D 均为用户 A 的直接关注者，代表了用户 A 的子网络中心用户。分析过程如图4-10所示。为了追踪新闻传播的动态过程，需要关注时间轴视图中新闻传播数量随时间的变化关系。由于传播总量和用户传播数量的差距较大，如果同时显示不便于观察变化趋势，因此本节将时间轴视图的三种不同比例尺同时显示在图4-10 T 中。根据初步观察，可以将总体传播过程划分为 7 个时间段 S1-S7，并分别截取各时间段分界时刻（分别为传播开始后 20 分钟、1 小时、2 小时、4 小时、6 小时、8 小时）的加权网络视图，如图4-10 N1-N6 所示。

观察图4-10 T 可知，在初始阶段新闻传播总量呈现缓慢增长（S1），之后快速增长一段时间（S2-S5），最终增长速率逐渐减慢并接近饱和（S6-S7），在快速增长期（S2-S5）中存在一段增长速率相对较慢的平台期（S4）。而对于用户 A、B、C、D 的传播数量，其增长趋势均与总量相似，但并未出现平台期。通过分析还可发现，用户 A 与用户 B、C、D 的快速增长时段之间存在错位，是总量增长平台期（S4）的产生原因。

具体来说，在 S1 时段，新闻传播总量的增长十分缓慢。观察图4-10 N1 可知，只有部分用户 A 的直接关注者被激活，且传播路径的颜色较浅，表明生成概率（指传播路径的生成概率，下同）均较低，没有间接关注者被激活。

在 S2-S3 时段，新闻传播总量进入第一个快速增长期，增量主要由用户 A 贡献。该时段同时也是用户 A 的快速增长期，结合图4-10 N2 和 N3 可以发现，用

户 A 的直接关注者均已被激活，并且生成概率在迅速增加。在传播开始后的 2 小时，用户 A 的直接关注者的传播路径颜色大多为橙色，表明生成概率大多已经达到约 0.5 左右。而对于间接关注者，在 S2 阶段开始有部分间接关注者被激活，在 S3 阶段则有更多的间接关注者被激活，但间接关注者的生成概率远低于直接关注者。此外，在观察过程中还注意到，用户 B 所在的子网络中，间接关注者被更早地激活，并且生成概率相比其它子网络更高。

在 S4 时段，总增长速率进入一个平台期，这主要是因为用户 A 的增长速率放缓，而其子网络中心用户仍处于缓慢增长状态。如图4-10 N4 所示，到传播开始后 4 小时，各个子网络中的生成概率都有不同程度的增长。

在 S5 时段，新闻传播总量进入第二个快速增长期，这是用户 A 的子网络中心用户开始迅速增长导致的，而用户 A 的增长速率仍在缓慢下降。如图4-10 N5 所示，在传播开始后 6 小时，直接关注者的传播路径颜色已经接近浅紫色，表明其生成概率已经接近约 0.8 及以上，间接关注者的生成概率则大幅度增长了。并且还可以注意到，用户 B 的重要程度明显高于除用户 A 外的其他用户。

在 S6 时段，总增长速率逐渐放缓，用户 A 的子网络中心用户的增长也逐渐放缓。观察图4-10 N6 可以发现，在传播开始后 8 小时，直接关注者的生成概率相比 6 小时只有少量增长，而间接关注者的生成概率增长幅度则更高。

最后，在 S7 阶段，总量和各个用户的增长进一步放缓，表示传播过程逐渐接近饱和。

4.4.3.3 任务三：分析新闻传播中存在的差异性

任务二的分析清晰地揭示了新闻传播过程中各个阶段的特点，但与任务一相同，在分析过程中同样发现新闻传播中存在差异性。本小节将分别针对差异性的三个具体方面，对三个典型现象的产生原因进行分析：

1. **直接关注者和间接关注者之间的差异性。**相比直接关注者，为什么间接关注者的最终生成概率更低？
2. **不同用户组之间的差异性。**为什么子网络中心用户都属于被标记为橙色的用户组，以及属于红色和橙色用户组的用户生成概率为什么大多高于属于蓝色和绿色用户组的生成概率？
3. **不同子网络之间的差异性。**为什么用户 B 比其它子网络中心用户更加重要？

要解答上述问题，需要在图4-10的基础上结合用户特征视图和用户分布视图进行分析，如图4-11所示。

对于问题 1，在直接关注者和间接关注者中分别选取两个典型用户 E 和 F，观察其传播路径的特征可以发现，由于前驱用户影响力的差异，间接关注者的传

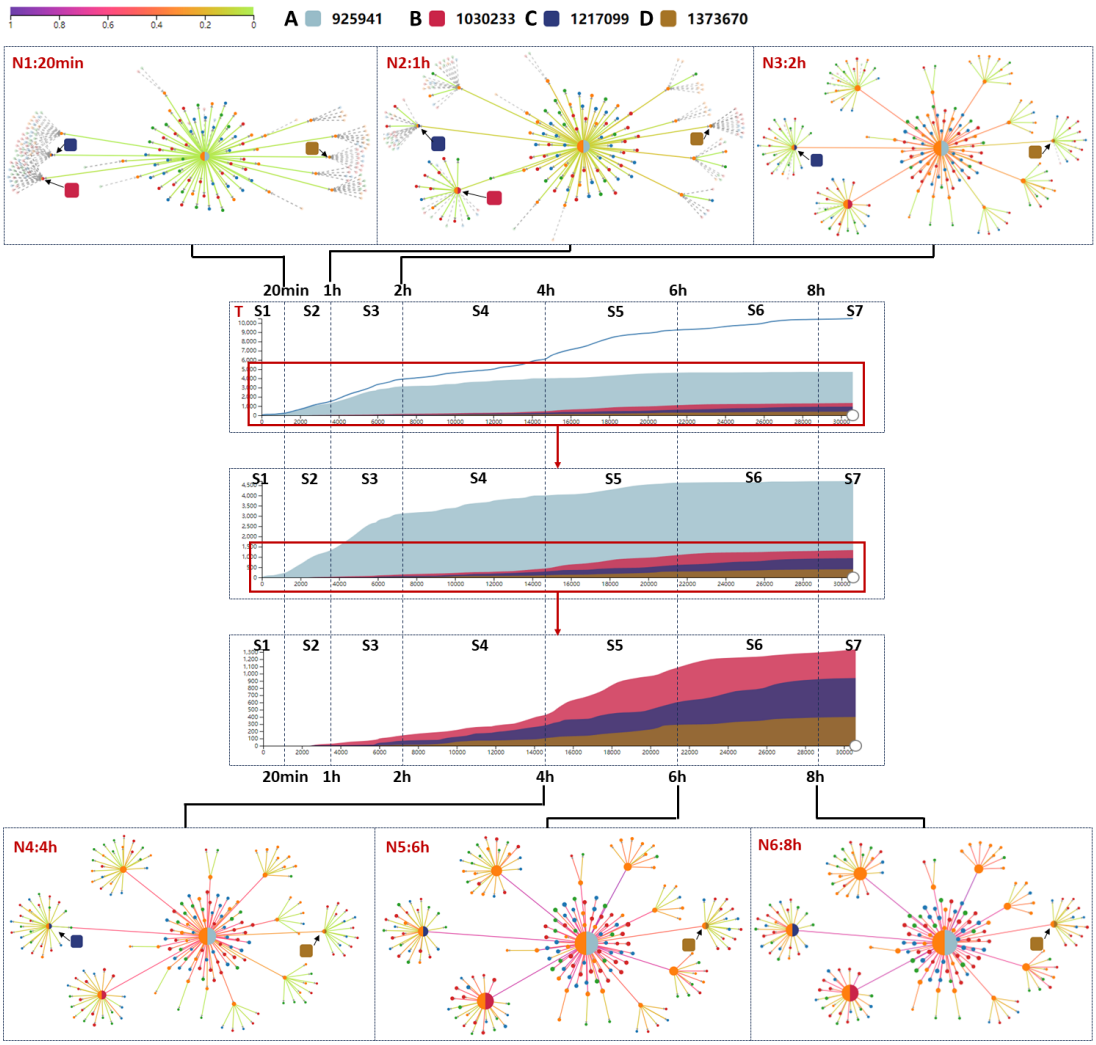


图 4-10　任务二：追踪新闻传播的动态过程

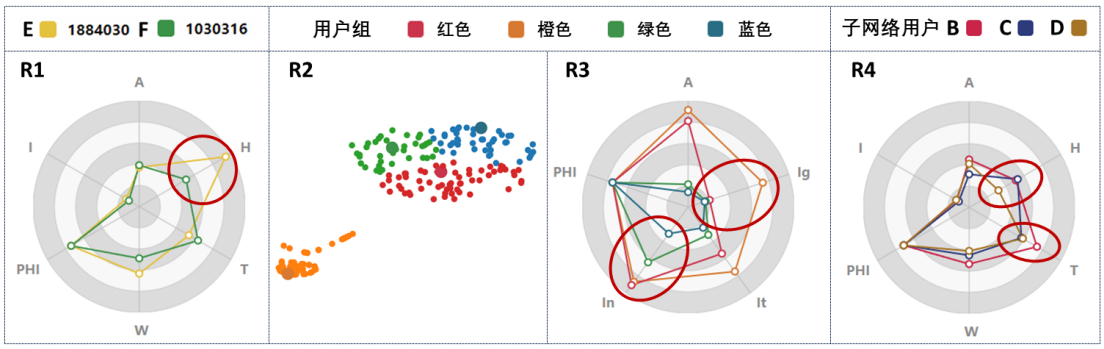


图 4-11　任务三：分析新闻传播中存在的差异性

播概率通常要比同一时刻未激活的直接关注者更低，因此它们的激活时间更晚，在流行度总量一定的前提下，大量流行度增量已经由直接关注者贡献，导致间接关注者的生成概率更低。

特别要说明的是，这里指出的间接关注者的激活时间更晚，并不是指间接关注者因其前驱用户未被激活，导致其不存在可激活的传播路径而无法被激活。而是指在间接关注者和直接关注者都处于可以被激活的状态时，间接关注者的激活时间仍然比直接关注者更晚。

具体到分析过程，用户 E（被标记为黄色）是用户 A 的某个直接关注者，用户 F（被标记为绿色）是用户 A 的某个间接关注者，并且处于较早被激活的用户 B 的子网络中。图4-11 R1 展示了传播开始后 20 分钟时上述用户的传播路径特征，在该时刻，上述两个用户都未被激活。观察特征可以发现，用户 E 的前驱用户影响力 H 要显著大于用户 F，而其它特征则较为相近，这是因为用户 E 的前驱用户 A 比用户 F 的前驱用户 B 在网络结构上更加重要。上述特征表明，由于用户 E 的前驱用户影响力更强，因此决定了用户 E 有更强的传播意愿，其传播概率更高，激活时间早于用户 F。由于用户 F 位于较早被激活的用户 B 的子网络中，那么位于其他更晚被激活的子网络中的间接关注者，其激活时间则通常更晚。而根据 SimCas 方法的定义，流行度的总增量是一定的，在间接关注者激活之前，大量流行度增量已经由直接关注者贡献，因此导致间接关注者最终的生成概率更低。

对于问题 2，在目前划分的四个用户组中，每组分别选取一个用户观察其用户特征，可以发现橙色用户通常拥有较高的影响力和活跃度，可能为拥有很多的粉丝且较为活跃的明星账号或官方账号，因此子网络的中心用户都属于橙色用户。而相比于蓝色用户和绿色用户，橙色和红色用户的活跃度更高，这是其生成概率通常更高的原因。

具体来说，图4-11 R2 展示了选择用户在二维平面中的投影位置，用户特征如图4-11 R3 所示。通过观察图中特征可以发现，橙色用户普遍具有较高的活跃度 A、全局影响力 I_g 和拓扑连通性 I_t，这表明橙色用户的活跃度很高并且在网络结构中的重要性较强，因此猜想橙色用户可能为拥有很多的粉丝且较为活跃的明星账号或官方账号。对于红色用户，虽然具有较高的活跃度 A，但其全局影响力 I_g 显著低于橙色用户，因此猜想红色用户可能为较为活跃的一般用户，它们虽然没有许多粉丝关注，在网络结构中的重要度不高，但活跃度很高，转发新闻的速度和频率较快。而绿色用户和蓝色用户，其各项特征均较低，表明他们是活跃程度一般并且没有太多粉丝关注的普通用户。同时，蓝色用户的虚假关注者得分 I_n 较低，说明蓝色用户为僵尸用户的可能性更高。因此，橙色用户和红色用户的传播概率通常要高于蓝色用户和绿色用户，使得它们的生成概率更高。

对于问题 3，通过对比用户 B 与用户 C、D 各自子网络中用户的传播路径特

征可以发现，与子网络规模相似的用户 C 相比，用户 B 子网络中的用户由于对其信任程度更高，因此激活时间更早，在流行度总量一定的前提下，最终平均生成概率更高，使得用户 B 的重要程度更高；而与用户 D 相比，用户 B 主要因为子网络规模更大，传播数量更多，因此重要程度更高。

具体来说，分别选取用户 B、C、D 子网络中的某一个用户（使用用户 B、C、D 对应的标注颜色进行表示），其中用户 C 代表了规模相似但重要程度更低的子网络，用户 D 则代表了规模更小的子网络。上述用户传播开始后 1 小时的传播路径特征如图4-11 R4 所示。分析图中特征可以发现，与用户 C 相比，用户 B 的子网络用户对其信任程度 T 更高。根据 3.2.3 章节中对信任程度的定义，这说明用户 B 子网络中的用户与其更加相似，从而使得传播意愿更强，传播概率更高。另一方面，与用户 D 相比，用户 B 的子网络用户在前驱用户影响力 H 方面明显更高，这表明用户 B 在网络结构上的重要程度更高，同样使得其传播概率更高。因此，用户 B 子网络中的用户能更早被激活，使得用户 B 的传播数量更多，重要程度更高。

4.4.3.4 应用讨论

通过上述分析任务可以总结出以下结论：

1. **新闻的传播不是均匀的。**在过程上呈现出明显的阶段性，在直接关注者和间接关注者之间、不同用户组之间、不同子网络之间的传播情况呈现差异性。
2. **新闻的传播过程存在阶段性规律。**在初始阶段呈现缓慢增长，之后则快速增长一段时间，最终增长速率逐渐减慢并接近饱和。并且在新闻的快速增长期中，存在一段增长速率相对较慢的平台期。
3. **新闻传播差异性的产生原因与用户和传播路径的特征相关。**间接关注者由于前驱用户影响力更低，使得生成概率低于直接关注者；拥有更高活跃度和全局影响力的用户通常是子网络的中心用户，并且生成概率更高；与其粉丝更相似的用户，其子网络内用户的平均生成概率更高。

上述结论可以为现实领域的应用提供指导。例如，对于希望新闻迅速传播的广告投放商，为了缩短增长平台期持续的时间，就需要在传播开始的早期对子网络中的间接关注者给予更多的流量扶持，使得它们能够被更早地激活，提前进入快速增长阶段；对于需要遏制假新闻传播的平台监管者，需要重点阻止活跃度更高的用户传播假新闻；对于要引导舆论的官方媒体，需要将资源重点倾斜至一些与粉丝相似度更高的关键用户，如用户 B，从而更好地引导舆论走向，向更多用户传播正能量内容。

4.5 本章小结

本章提出了一种基于流行度的在线社交网络新闻传播过程分析方法 SimCas。SimCas 方法首先对新闻传播过程进行模拟，提出了一个带流行度约束的新闻传播过程模拟算法 PopSim，使用流行度预测结果约束模拟过程，得到一次模拟的新闻传播网络及其生成概率。SimCas 方法使用 PopSim 算法进行多次模拟，并设计了一个概率加权新闻传播网络生成算法 ProNet，将多次模拟结果聚合为概率加权新闻传播网络。之后，SimCas 方法根据模拟结果，使用一个可视分析系统 SimVis 进行分析。本章通过真实数据集上的实验验证了 SimCas 方法模拟结果的准确性，并通过案例分析验证了其分析功能的有效性。

第 5 章 总结与展望

5.1 本文工作总结

针对在线社交网络上的新闻传播分析问题，本文首先提出了一个基于传播意愿的在线社交网络新闻流行度预测模型 WillCas，使用一个注意力图神经网络进行实现，并通过对比实验和消融实验验证了该模型；随后本文提出了一种基于流行度的在线社交网络新闻传播过程分析方法 SimCas，使用一个带流行度约束的新闻传播过程模拟算法 PopSim 和一个概率加权传播网络生成算法 ProNet 对新闻传播过程进行模拟。在验证了模拟结果的准确性后，SimCas 方法使用一个可视分析系统 SimVis 进行分析，并通过案例验证了分析功能的有效性。

本文提出的基于传播意愿的在线社交网络新闻流行度预测模型 WillCas 综合考虑了用户传播意愿、时间特征、网络结构特征三个方面的影响因素。WillCas 模型从用户活跃程度、前驱用户影响力、基于用户画像的信任程度三个方面来计算用户传播意愿；使用幂律分布建模时间特征；通过全局影响力、拓扑连通性和虚假关注者得分三个角度构建用户网络特征。本文使用一个端到端的注意力图神经网络框架实现了 WillCas 模型，该框架通过随机游走的方法采样传播网络，使用双向 GRU 学习隐藏表示，通过注意力机制优化预测结果。本文在真实数据集上进行了对比实验和消融实验，实验结果表明，WillCas 模型可以较为准确和稳定地预测在线社交网络的新闻流行度，其各部分特征具有有效性。

本文提出的基于流行度的在线社交网络新闻传播过程分析方法 SimCas 首先对新闻传播过程进行模拟。SimCas 方法提出了一种带流行度约束的新闻传播过程模拟算法 PopSim，PopSim 算法使用 WillCas 模型预测的流行度约束模拟过程，生成一种新闻传播网络，并计算了生成概率。为了消除 PopSim 算法的随机性，探索传播过程的一般性规律，SimCas 方法使用 PopSim 算法进行多次模拟，并设计了一个概率加权新闻传播网络生成算法 ProNet，将多次模拟结果聚合为一个概率加权新闻传播网络。之后，SimCas 方法根据模拟结果，使用一个可视分析系统 SimVis 对新闻传播过程进行分析。本文通过真实数据集上的实验验证了 SimCas 方法的模拟结果具有较好的准确性，并通过案例分析的方式进一步评估了其分析功能的有效性。

5.2 未来工作展望

本文提出的 WillCas 模型和 SimCas 方法虽然都具有较好的准确性和有效性，但也存在着一些不足和改进之处，主要包括以下几个方面：

1. 用户兴趣程度也应作为计算用户传播意愿时的重要因素。新闻可以按主题进行分类，用户兴趣程度体现了用户在哪些类型的新闻内容上更感兴趣，显然用户会更愿意传播自己更感兴趣的新闻内容。但目前同时拥有传播网络数据和文本内容数据的公开数据集数量较少、质量较差，同时社交网络平台的用户隐私政策也限制了对文本内容的直接爬取，因此本文提出的 WillCas 模型中没有考虑用户兴趣程度。在未来的工作中，探索在隐私政策允许下构建高质量数据集、计算用户兴趣程度的方法仍是一个挑战。
2. 在本文提出的 SimCas 方法中，PopSim 算法假设了用户只能被一条传播路径激活。但在真实情况下，用户可能会从多个前驱用户处转发相同的内容，甚至用户会转发自己发布过的内容。同时，PopSim 算法仅适用于较小规模的用户网络，在面对大规模用户网络时，计算条件传播概率仍会导致组合数爆炸等问题，影响算法性能，本文中提出的近似计算和剪枝聚类等方法仍不能很好地解决问题。因此，进一步优化 PopSim 算法，使其更加符合真实情况、降低算法复杂度也是未来工作的改进方向之一。

参考文献

[1] Cao Q, Shen H, Cen K. DeepHawkes: Bridging the Gap between Prediction and Understanding of Information Cascades [C]. 2017: 1149-1158.

[2] Chen X, F Z, K Z. Information Diffusion Prediction via Recurrent Cascades Convolution [C]. IEEE, 2019: 770-781.

[3] Wu L, C, Hsieh P, H, J J. Muffle: Multi-modal fake news influence estimator on twitter [J]. 2019 IEEE 35th international conference on data engineering (ICDE), Applied Sciences, 2022, 12(1): 453.

[4] Tatar A, De Amorim M D, Fdida S. A survey on predicting the popularity of web content [J]. Journal of Internet Services and Applications, 2014, 5(1): 1-20.

[5] Zhong C, Xiong F, Pan S. Hierarchical attention neural network for information cascade prediction [J]. Information Sciences, 2023, 622: 1109-1127.

[6] Cerchiello P, Giudici P, Nicola G. Twitter data models for bank risk contagion [J]. Neurocomputing, 2017, 264: 50-56.

[7] Li C, Ma J, Guo X. Deepcas: An end-to-end predictor of information cascades [C]. 2017: 577-586.

[8] Zhou F, Xu X, Trajcevski G. A survey of information cascade analysis: Models, predictions, and recent advances [J]. ACM Computing Surveys (CSUR), 2021, 54(2): 1-36.

[9] Tsugawa S. Empirical analysis of the relation between community structure and cascading retweet diffusion [C]. 2019, 13: 493-504.

[10] Quattrociocchi W, Scala A, Sunstein C R. Echo chambers on Facebook [J]. Available at SSRN 2795110, 2016.

[11] Xiao C, Liu C, Ma Y. Time sensitivity-based popularity prediction for online promotion on Twitter [J]. Information Sciences, 2020, 525: 82-92.

[12] Kupavskii A, Ostroumova L, Umnov A, et al. Prediction of retweet cascade size over time [C]. In Proceedings of the 21st ACM international conference on Information and knowledge management, 2012: 2335–2338.

[13] Yang C, Sun M, Liu H. Neural diffusion model for microscopic cascade study [J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 33(3): 1128-1139.

[14] Wang J, Zheng V W, Liu Z. Topological recurrent neural network for diffusion prediction [C]. 2017: 475-484.

[15] Weng L, Menczer F, Ahn Y Y. Virality prediction and community structure in social networks [J]. Scientific reports, 2013, 3(1): 1-6.

[16] Mcauley J J, Leskovec J. Learning to discover social circles in ego networks [C]. 2012, 25.

[17] Szabo G, Huberman B A. Predicting the popularity of online content [J]. Communications of the ACM, 2010, 53(8): 80-88.

[18] Bao P, Shen H W, Huang J. Popularity prediction in microblogging network: a case study on sina weibo [C]. 2013: 177-178.

[19] J C, Adamic L, Dow P A. Can cascades be predicted? [C]. 2014: 925-936.

[20] Zaman T, Fox E B, Bradlow E T. A Bayesian approach for predicting the popularity of tweets [J]. Institute of Mathematical Statistics, 2014(3).

[21] Cui P, Jin S, Yu L. Cascading outbreak prediction in networks: a data-driven approach [C]. 2013: 901-909.

[22] Gao S, Ma J, Chen Z. Popularity prediction in microblogging network [C]. 2014: 379-390.

[23] Wu B, Shen H. Analyzing and predicting news popularity on Twitter [J]. International Journal of Information Management, 2015, 35(6): 702-711.

[24] Petrovic S, Osborne M, Lavrenko V. Rt to win! predicting message propagation in twitter [C]. 2011, 5(1): 586-589.

[25] Suh B, Hong L, Pirolli P. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network [C]. IEEE, 2010: 177-184.

[26] Jenders M, Kasneci G, Naumann F. Analyzing and predicting viral tweets [C]. 2013: 657-664.

[27] Tatar A, Antoniadis P, Amorim M D. From popularity prediction to ranking online news [J]. Social Network Analysis and Mining, 2014, 4: 1-12.

[28] Hong L, Dan O, Davison B D. Predicting popular messages in twitter [C]. 2011: 57-58.

[29] Goel S, Anderson A, Hofman J. The structural virality of online diffusion [J]. Management Science, 2016, 62(1): 180-196.

[30] Jamali S, Rangwala H. Digging digg: Comment mining, popularity prediction, and social network analysis [C]. IEEE, 2009: 32-38.

[31] Dong Y, Johnson R A, Chawla N V. Will this paper increase your h-index? Scientific impact prediction [C]. 2015: 149-158.

[32] Bandari R, Asur S, Huberman B. The pulse of news in social media: Forecasting popularity [C]. 2012, 6(1): 26-33.

[33] Khabiri E, Hsu C F, Caverlee J. Analyzing and predicting community preference of socially generated metadata: A case study on comments in the digg community [C]. 2009, 3(1): 238-241.

[34] Yuan N J, Zhong Y, Zhang F. Who will reply to/retweet this tweet? The dynamics of intimacy from online social interactions [C]. 2016: 3-12.

[35] Gao S, Ma J, Chen Z. Modeling and predicting retweeting dynamics on microblogging platforms [C]. 2015: 107-116.

[36] Shen H, Wang D, Song C. Modeling and predicting popularity dynamics via reinforced poisson processes [C]. 2014, 28(1).

[37] Lu X, Yu Z, Guo B. Predicting the content dissemination trends by repost behavior modeling in mobile social networks [J]. Journal of Network and Computer Applications, 2014, 42: 197-207.

[38] Zadeh A H, Sharda R. Modeling brand post popularity dynamics in online social networks [J]. Decision Support Systems, 2014, 65: 59-68.

[39] Mishra S, Rizoiu M A, Xie L. Feature driven and point process approaches for popularity prediction [C]. 2016: 1069-1078.

[40] Ding W, Shang Y, Guo L. Video popularity prediction by sentiment propagation via implicit network [C]. 2015: 1621-1630.

[41] 丁学君. 基于 SCIR 的微博舆情话题传播模型研究 [J]. 计算机工程与应用, 2015 (8): 20−26.

[42] Zhao Q, Erdogdu M A, He H Y. Seismic: A self-exciting point process model for predicting tweet popularity [C]. 2015: 1513-1522.

[43] Li Q, Wu Z, Yi L. WeSeer: Visual analysis for better information cascade prediction of WeChat articles [J]. IEEE Transactions on Visualization and Computer Graphics, 2018, 26(2): 1399-1412.

[44] Yu L, Cui P, Wang F. Uncovering and predicting the dynamic process of information cascades with survival model [J]. Knowledge and information systems, 2017, 50: 633-659.

[45] Li D, Xu Z, Luo Y, et al. Modeling information diffusion over social networks for temporal dynamic prediction [C]. In Proceedings of the 22nd ACM international conference on Information & Knowledge Management, 2013: 1477–1480.

[46] Guan X, Peng Q, Li Y. Hierarchical neural network for online news popularity prediction [C]. IEEE, 2017: 3005-3009.

[47] Saeed R, Abbas H, Asif S. A framework to predict early news popularity using deep temporal propagation patterns [J]. Expert Systems with Applications, 2022, 195: 116496.

[48] Dou H, Zhao W X, Zhao Y. Predicting the popularity of online content with knowledge-enhanced neural networks [C]. 2018.

[49] Cao Q, Shen H, Gao J. Popularity prediction on social platforms with coupled graph neural networks [C]. 2020: 70-78.

[50] Wang Y, Wang X, Ran Y, et al. CasSeqGCN: Combining network structure and temporal sequence to predict information cascades [J]. Expert Systems with Applications, 2022, 206: 117693.

[51] Ding Y, Wang B, Cui X. Popularity prediction with semantic retrieval for news recommendation [J]. Expert Systems with Applications, 2024: 123308.

[52] Qi T, Wu F, Wu C, et al. Pp-rec: News recommendation with personalized user interest and time-aware news popularity [J]. arXiv preprint arXiv:2106.01300, 2021.

[53] Huang Z, Wang Z, Zhang R, et al. Learning Bi-directional Social Influence in Information Cascades using Graph Sequence Attention Networks [C]. In WWW '20: The Web Conference 2020, 2020.

[54] Sun X, Zhou J, Liu L, et al. Explicit time embedding based cascade attention network for information popularity prediction [J]. Information Processing Management: Libraries and Information Retrieval Systems and Communication Networks: An International Journal, 2023.

[55] Wang Y, Wang X, Jia T. CCasGNN: Collaborative Cascade Prediction Based on Graph Neural Networks [J], 2021.

[56] Wicaksono A S, Supianto A A. Hyper parameter optimization using genetic algorithm on machine learning methods for online news popularity prediction [J]. International Journal of Advanced Computer Science and Applications, 2018, 9(12).

[57] Yang，C, Wang H, Tang J. Full-scale information diffusion prediction with reinforced recurrent networks [J]. IEEE Transactions on Neural Networks and Learning Systems, 2021.

[58] Sanjo S, Katsurai M. Recipe popularity prediction with deep visual-semantic fusion [C]. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017: 2279–2282.

[59] Mayer R C, Davis J H, Schoorman F D. An integrative model of organizational trust [J]. Academy of management review, 1995, 20(3): 709-734.

[60] Ghafari S M, Yakhchi S, Beheshti A. SETTRUST: social exchange theory based context-aware trust prediction in online social networks [C]. 2019: 46-61.

[61] Ghafari S M, Beheshti A, Joshi A. A survey on trust prediction in online social networks [J]. IEEE Access, 2020, 8: 144292-144309.

[62] Golbeck J. Trust and nuanced profile similarity in online social networks [J]. ACM Transactions on the Web (TWEB), 2009, 3(4): 1-33.

[63] Tang J, Gao H, Hu X. Exploiting homophily effect for trust prediction [C]. 2013: 53-62.

[64] Abbasi M A, Liu H. Measuring user credibility in social media [C]. 2013: 441-448.

[65] Hang C W, Singh M P. Trust-based recommendation based on graph similarity [C]. 2010, 82.

[66] Adali S, Escriva R, Goldberg M K. Measuring behavioral trust in social networks [C]. IEEE, 2010: 150-152.

[67] Liu G, Liu Y, Liu A, et al. Context-aware trust network extraction in large-scale trust-oriented social networks [J]. World Wide Web, 2018, 21: 713–738.

[68] Ghafari S M, Yakhchi S, Beheshti A. Social context-aware trust prediction: methods for identifying fake new [C]. 2018: 161-177.

[69] Chen S, Chen S, Lin L. E-map: A visual analytics approach for exploring significant event evolutions in social media [C]. IEEE, 2017: 36-47.

[70] Chen S, Chen S, Wang Z. D-map+ interactive visual analysis and exploration of ego-centric and event-centric information diffusion patterns in social media [J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2018, 10(1): 1-26.

[71] Chen S, Li S, Chen S. R-map: A map metaphor for visualizing information reposting process in social media [J]. IEEE transactions on visualization and computer graphics, 2019, 26(1): 1204-1214.

[72] Cao N, Lin Y R, Sun X. Whisper: Tracing the spatiotemporal process of information diffusion in real time [J]. IEEE transactions on visualization and computer graphics, 2012, 18(12): 2649-2658.

[73] Ren D, Zhang X, Wang Z, et al. Weiboevents: A crowd sourcing weibo visual analytic system [C]. In 2014 IEEE Pacific Visualization Symposium, 2014: 330–334.

[74] Liu Y, Wang C, Ye P, et al. Analysis of micro-blog diffusion using a dynamic fluid model [J]. Journal of Visualization, 2015, 18: 201–219.

[75] Barnes J A. Class and committees in a Norwegian island parish [J]. Human relations, 1954, 7 (1): 39–58.

[76] Schneider F, Feldmann A, Krishnamurthy B, et al. Understanding online social network usage from a network perspective [C]. In Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement, 2009: 35–48.

[77] Watts D J, Strogatz S H. Collective dynamics of ‘small-world’ networks [J]. nature, 1998, 393 (6684): 440–442.

[78] Backstrom L, Huttenlocher D, Kleinberg J, et al. Group formation in large social networks: membership, growth, and evolution [C]. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, 2006: 44–54.

[79] Jeh G, Widom J. Simrank: a measure of structural-context similarity [C]. 2002: 538-543.

[80] Fogaras D, Rácz B. Scaling link-based similarity search [C]. In Proceedings of the 14th international conference on World Wide Web, 2005: 641–650.

[81] Page L. The pagerank citation ranking [J]. http://www-db. stanford. edu/backrub/pageranksub. ps, 1998.

[82] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [J]. arXiv preprint arXiv:1406.1078, 2014.

[83] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate [J]. arXiv preprint arXiv:1409.0473, 2014.

[84] Qiu J, Tang J, Ma H. Deepinf: Social influence prediction with deep learning [C]. 2018: 2110-2119.

[85] Cao R, Geng Y, Xu X, et al. How does duplicate tweeting boost social media exposure to scholarly articles? [J]. Journal of Informetrics, 2022, 16 (1): 101249.

[86] Enders A, Hungenberg H, Denker H P. The long tail of social networking.: Revenue models of social networking sites [J]. European Management Journal, 2008, 26(3): 199-211.

[87] Wikipedia contributors. Ghost followers — Wikipedia, The Free Encyclopedia. 2024. https://en.wikipedia.org/w/index.php?title=Ghost_followers&oldid=1201771410[Online; accessed 16-February-2024].

[88] Sankar A, Zhang X, Krishnan A. Inf-VAE: A variational autoencoder framework to integrate homophily and influence in diffusion prediction [C]. 2020: 510-518.

[89] Han S, Zhuang F, He Q, et al. Energy model for rumor propagation on social networks [J]. Physica A: Statistical Mechanics and its Applications, 2014, 394: 99–109.

[90] Litou I, Kalogeraki V, Katakis I, et al. Real-time and cost-effective limitation of misinformation propagation [C]. In 2016 17th IEEE international conference on mobile data management (MDM), 2016: 158–163.

[91] Zhang J, Liu B, Tang J, et al. Social influence locality for modeling retweeting behaviors [C]. In Twenty-third international joint conference on artificial intelligence, 2013.

[92] Sanjo S, Katsurai M. Recipe popularity prediction with deep visual-semantic fusion [C]. 2017: 2279-2282.

[93] Roweis S T, Saul L K. Nonlinear dimensionality reduction by locally linear embedding [J]. science, 2000, 290 (5500): 2323–2326.

[94] Van der Maaten L, Hinton G. Visualizing data using t-SNE. [J]. Journal of machine learning research, 2008, 9 (11).

发表论文和参加科研情况说明

（一）申请及已获得的专利

[1] 张怡，刘兴宇，李会彬. 一种基于轨迹分割的渲染路径预测方法（申请中）：中国，202310273548.2[P]. 2023-03-20.

[2] 张怡，刘兴宇. 一种基于传播意愿的话题流行度预测模型（申请中）：中国，202410617600.6[P]. 2024-05-17.

（二）参与的科研项目

[1] “文物知识组织表达模型与标准规范研究”，科技部国家重点研发计划子课题 (No.2019YFC1521202)，2020.1-2023.12

[2] “大型综合性博物馆全生命周期数字孪生理论模型研究”，国家重点研发计划课题 (No.2022YFF0904301)，2022.11-2025.10

致　　谢

时光飞逝，我的研究生生活即将结束。经过三年的学习，我最终完成了这篇硕士学位论文，我感到非常高兴。在此，我要感谢所有给予过我帮助和支持的人。

首先，我要感谢我的导师张怡老师，您的悉心教导和丰富经验使我受益匪浅。在选择课题时，张老师帮助我细化研究方向，分析技术路线；在遇到困难时，张老师总是耐心指导，为我提出解决方案；在论文遇到问题时，张老师鼓励我不要灰心，手把手帮助我修改论文。三年里，张老师始终支持我的学习和生活，包容我的缺点与过失，您的关心和帮助如同家人一般温暖，您严谨治学、亲切温柔的品格深刻影响了我。我还要感谢李罡老师在横向项目中对我的帮助，从本科开始，您专业严谨的工作风格始终是我学习的榜样；感谢陈锦言老师为我提供计算资源；感谢辅导员孙媛老师对我的帮助和照顾。

其次，我要感谢父母和家人。他们不仅在经济上资助我完成学业，还始终关心着我的学习生活，帮助我排忧解难，是我完成学业的坚实后盾。

之后，我要感谢我的同学和朋友们。感谢李朝晖学长向我分享知识、指引研究方向；感谢李思思、董浩天同学帮助我修改论文；感谢刘杭学、朱林刚同学在论文、横向项目等工作中提供的帮助；感谢马辰、陈浩阳、于海新、郭云飞等同学在实验室生活中对我的帮助；感谢室友王昊宇同学在生活中对我的照顾。

我还要特别感谢几位朋友，在最艰难的那些日子里，是他们的陪伴支撑着我渡过难关。感谢我的高中同学吴非、卢纯青，他们经常关心我的学习生活，陪我聊天打游戏，缓解我的压力；感谢我的本科室友孙浩铭，他经常与我聊天，探讨各类问题，充实了我的课余生活；感谢本科时在社团认识的同学郑瑞凡，虽然不属于同一个学院，但他丰富的科研经验为我撰写论文提供了许多帮助，不仅如此，我们还经常一起吃饭、聊天、玩手办，这段持续了六年的友谊弥足珍贵。

在过去的一年中，我面临毕业和就业的双重压力，一边备考一边完成研究课题，经历过多次的情绪波动。但好在我没有放弃，克服了诸多困难，闯过了一道道难关，最终顺利完成了学业。我要感谢持续坚持努力的自己，也要感谢头顶璀璨的星空、耳边流淌的音乐、二次元世界的企划和众多同好，是它们给了我源源不断的前进动力。

三年的研究生生涯中，我不仅学到了专业知识，提升了综合能力，还体验了许多新鲜事物，培养了兴趣爱好，收获了很多成长与快乐。虽然经历了不少困难与曲折，但这段经历已经成为我人生中最宝贵的回忆。