

# 杭州师范大学

## 本科生毕业设计（论文） (2019 届)

题 目 复杂网络理论及其应用

学 院 理学院

专 业 信息与计算科学

班 级 信息 151

学生姓名 许宇星 学 号 2015210202018

指导教师 李安水 职 称 讲师

完成日期 2019 年 4 月 30 日

答辩日期 2019 年 5 月 8 日

成 绩

# **复杂网络理论及其应用**

**Some aspects of complex network theory and its  
applications**

**学    院：理学院**

**专    业：信息与计算科学**

**班    级：信息 151**

**学    号：2015210202018**

**学生姓名：许宇星**

**指导教师：李安水**

**2019 年 4 月 30 日**

---

## 摘 要

*Facebook*、*Twitter*、微信和微博等社交和信息网络活动已经成为我们日常生活中不可或缺的一部分，在那里，我们可以很容易地接触到朋友的行为，并进而受到朋友的影响。因此，对每个用户进行有效的社会影响预测对于各种应用程序至关重要。传统的社会影响预测方法通常设计各种手工规则来提取特定于用户和网络的特征。然而，它们的有效性在很大程度上取决于领域专家的知识。因此，通常很难将它们推广到不同的领域。

在社交相似性和 *GHSOM* 的启发下，本文研究了图特征以及手工特征输入的基于 *GHSOM* 的社交影响力算法，学习用户的潜在特征表示，以预测社会影响。将网络结构和特定用户特征在卷积神经网络中补齐。*GHSOM* 将用户的特征作为输出到一个二维的表征网络，以表现其潜在的社会表征。在微博上进行的广泛实验表明，所提出的基于 *GHSOM* 的新模型显著优于传统的基于特征工程的方法。

**关键词：**图卷积;*GHSOM*; 社交影响力; 社交网络

---

## Abstract

Social and information networking activities such as on Facebook, Twitter, WeChat, and Weibo have become an indispensable part of our everyday life, where we can easily access friends' behaviors and are in turn influenced by them. Consequently, an effective social influence prediction for each user is critical for a variety of applications such as online recommendation and advertising.

Conventional social influence prediction approaches typically design various hand-crafted rules to extract user- and network- specific features. However, their effectiveness heavily relies on the knowledge of domain experts. As a result, it is usually difficult to generalize them into different domains. Inspired by social similarity and GHSOM, We design a social influence algorithm based on GHSOM, which has both graph-fixed-point feature and manual feature input. We can learn the representation of users' potential features to predict the social impact. The structure of the network and the user-specific features are complemented in the convolution neural network. GHSOM outputs the user's features into a two-dimensional representation network to represent its potential social representation. Extensive experiments on Weibo show that the proposed GHSOM-based model is significantly superior to the traditional feature-based engineering method.

**Keywords:** Graph Convolution; Growing Hierarchical Self-Organizing Maps; Social Influence; Social Networks

## 目 录

第 1 章 绪论 .....	1
1.1 研究现状 .....	1
1.2 问题特征 .....	2
1.2.1 顶点特征 .....	2
1.2.2 手工特征 .....	3
第 2 章 主要模型 .....	4
2.1 GCN 模型 .....	4
2.2 GHSOM 模型 .....	5
第 3 章 实验设置 .....	6
3.1 数据集 .....	6
3.2 评估指标 .....	8
3.3 算法比较 .....	8
第 4 章 实验结果 .....	9
4.1 参数调整 .....	9
4.2 算法应用 .....	10
4.2.1 舆情监控 .....	10
4.2.2 广告投放 .....	10
第 5 章 相关工作 .....	11
第 6 章 结论 .....	12
第 7 章 使用简介 .....	13
7.1 数学公式、图表、参考文献等功能 .....	13
7.1.1 数学公式 .....	13
7.1.2 表格 .....	13
7.1.3 图片插入 .....	13
7.1.4 算法 .....	13
7.1.5 参考文献引用 .....	13
7.2 常见使用问题 .....	15
附录 .....	18
参考文献 .....	20
致 谢 .....	21

## 第 1 章 绪论

社会影响无处不在，不仅存在于我们的日常生活中，也存在于虚拟网络空间中。“社会影响”一词通常指一个人的情绪、观点或行为受到他人影响的现象。随着在线和移动社交平台的全球普及，人们已经看到了社会影响在各个领域的影响，如总统选举<sup>[1]</sup>、广告<sup>[2]</sup>等等。到目前为止，毫无疑问，社会影响已经成为推动我们社会决策的一种普遍而复杂的力量，明确需要方法来描述、理解和量化社会影响的基本机制和动态。

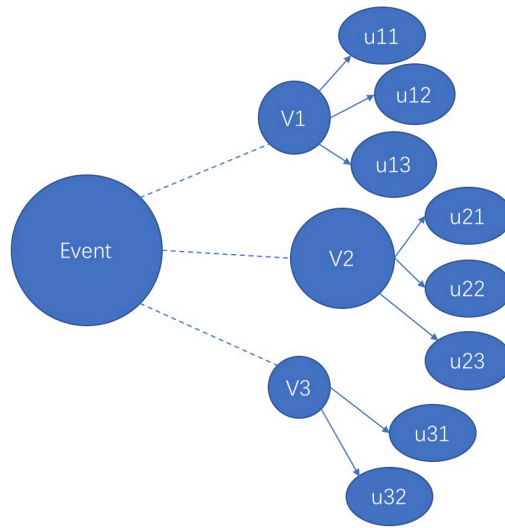


图 1.1 热点事件传播示意图

### 1.1 研究现状

事实上，许多文献都对社会影响预测做了大量的工作。例如，Matsubara 等人<sup>[3]</sup>通过精心设计从经典的“易感染”模型扩展而来的微分方程，研究了社会影响的动力学。最近，Li 等人<sup>[4]</sup>出了一种结合递归神经网络 (RNN) 和表示学习来推断级联大小的端到端预测方法。所有这些方法的主要目的是预测社会影响的全球或聚集模式，如在一个时间框架内的级联规模。然而，在许多在线应用中，如广告和推荐，有效地预测每个人的社会影响是至关重要的，即用户层面的社会影响预测。像 Twitter 这样的西方在线社交网络已经得到了很好的研究，但中国人气颇高的微博网络新浪微博的特点却一直很少被研究过。文献<sup>[5]</sup>发现，转发在新浪微博上更加普遍，并为创造趋势做出了很大贡献。

国内文献<sup>[7][8][9][10]</sup>尝试对于微博社交影响力进行多个角度的研究。其中文献<sup>[7]</sup>提出将粒子群算法应用于热点话题的发现，得出结果表明用户的转发操作对话题的影响力有比较大的影响的结论。文献<sup>[8]</sup>考虑用户话题信息传播能力以及用户与背景话题间关联性，提出了 TSRank 算法，但是上述论文都没有考虑使用者在整个社群网络结构中的各种重要程度。

本文主要研究用户层面的社会影响预测。我们的目的是通过某一热点话题在转发过程中不同用户的行动状态，用户的近邻的动作状态和该话题被转发的结构信息，给出该热点话题的影响力。例如，在图1.1中，当热点事件发生之后，常常是社区中的意见领袖刷线发布原创内容，经过普通用户层层传递，持续活跃直至周期结束。

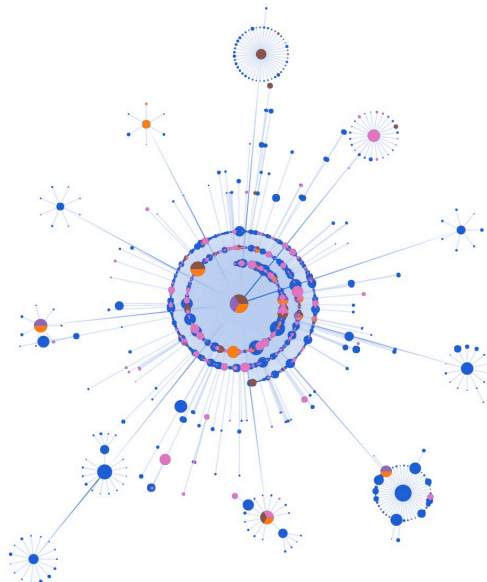


图 1.2 社交用户转发关系图 (不同的颜色表示不同关键词)

受到社交相似性在微博转发中的预测成功的启发<sup>[6]</sup>，我改进了基于 GHSOM<sup>[2]</sup> 算法，发现社会影响中隐藏的和预测的信号。通过将网络嵌入<sup>[2]</sup>、图卷积<sup>[7]</sup> 和 SOM 神经网络构建到一个方案中。在获得如图1.2所示的转发网络后，我们利用图卷积技术来补齐关键特征。然后经过 GHSOM 神经网络将高维特征中隐藏的信息映射到二维网络上，结合网络特征和自定义的手工特征计算社会影响力。经过广泛的实验，结果表明，模型可以方法能准确及有效评估微博用户的影响力。

## 1.2 问题特征

### 1.2.1 顶点特征

**定义 1.2.1.1 (度中心性)** 连结度中心性为该节点与其邻居节点的所有连结数的总合。

如式 (1.1) 所示。

$$C_D(n_i) = degree_{in}(n_i) + degree_{out}(n_i) \quad (1.1)$$

式 (1.1) 中， $degree_{in}(n_i)$  代表其他使用者连到  $n_i$  使用者的连结数， $degree_{out}(n_i)$  为使用者  $n_i$  连到其他使用者的连结数。

**定义 1.2.1.2 (邻近度中心性)** 邻近度中心性则是以该节点与其余所有在同一网络中的节点的距离总和来做计算。

$$C_c(n_i) = \frac{1}{\sum_{k \neq i} distance(n_i, n_k)} \quad (1.2)$$

如式 (1.2) 所示。式 (1.2) 中,  $distance(n_i, n_j)$  为  $n_i$  到  $n_j$  节点的距离。

**定义 1.2.1.3 (中介度中心性)** 中介度中心性则是用来计算除了该节点外的所有节点中, 任两节点之间的路径中, 通过该节点的路径数除以两节点所有路径数比值的总合。

当节点在网络中扮演着链接两个原先互不相连的节点之角色时, 该节点中介度中心性的值则会较高,

$$C_B(n_i) = \sum_{j, k \neq i} \frac{|path_{vian_i}(n_i, n_k)|}{|path_{total}(n_j, n_k)|} \quad (1.3)$$

如式 (1.3) 所示。式 (1.3) 中,  $path_{vian_i}(n_i, n_k)$  为节点  $n_j$  通过节点  $n_i$  而连结到节点  $n_k$  的最短路径数,  $path_{total}(n_j, n_k)$  则为节点  $n_j$  连结到节点  $n_k$  的所有最短路径数。

## 1.2.2 手工特征

**Repost influence** 则是用户本月所发布的信息被转发次数的平均, 该项指标能反映出使用者所发布之言论的价值。式 (1.4) 中,  $N_m$  为使用者每月所发布之信息量, 而  $\sum N_R$  则为每月每则信息被转发次数之总和。

$$I_r = 1 - \sqrt{\frac{N_m}{N_m + \sum N_R}} \quad (1.4)$$

**Repost depth** 转发深度, 用章节??中的被转发的层数的最大值来评估。

**Mention influence** 为用户每月所发布之信息被回复及评论的次数, 该项指标反映出使用者能吸引多少追随者来参与对话的能力, 也就是言论话题性。式 (1.5) 中,  $N_M$  为使用者每月所发布之信息量,  $\sum N_M$  则为每月每则信息被讨论次数之总和。

$$I_m = 1 - \sqrt{\frac{N_m}{N_m + \sum N_M}} \quad (1.5)$$

**Geographical influence** 地理影响分布, 参考[新浪微博的省份城市编码](#), 比如, 宁夏的省份编号是 64, 其下的主要城市分别为 1 银川、2 石嘴山、3 吴忠、4 固原, 青海的省份编号是 63, 其下主要城市有 8 个。我们发现地理上相邻的省份及城市, 其编号也非常接近, 我们建立影响广度特征公式 (1.6)。其中  $P$  表示省份,  $C$  表示城市,  $I_{i,j}$  表示在省份  $P$  下的城市  $C$  被影响的个数。

$$I_g = \frac{1}{P} \sum_{i \in P} \sum_{j \in C} \frac{I_{i,j}}{I_{i,total}} \quad (1.6)$$

检验预测结果的方法。**Spearman's rank correlation coefficients** 是用来分析两种排名算法之间相关性的量测方法。如式 (1.7) 所示。该分析方法是根据等级数据研究两个变量间相关性的方法, 相关性越高其分析结果越接近 +1, 相关性越低其分析结果越接近 -1。

$$\rho = 1 - \sqrt{\frac{6 \sum (x_i - y_i)^2}{N(N^2 - 1)}} \quad (1.7)$$



## 第2章 主要模型

这一部分将会依次介绍 GCN<sup>[7]</sup>、GHSOM<sup>[21]</sup>、特征量化模型，为下一节实验做准备。我将经过 GCN 处理的数据作为 GHSOM 的输入，将高维特征映射到二维空间，结合特征量化模型输出影响力数值。将影响力排名与微博话题榜做比较，使得斯皮尔曼的等级相关系数最大化。

### 2.1 GCN 模型

图卷积神经网络是一种用于图形结构数据的半监督学习算法。通过叠加多个 GCN 层建立 GCN 模型。每个 GCN 层的输入是一个顶点特征矩阵， $H \in \mathbb{R}^{n \times F}$ ，其中  $n$  是顶点数， $F$  是特征数，每行  $H$ ，用  $h_i^T$  表示，与一个顶点相关联。一般来说，GCN 层的本质是一个非线性变换，输出如下  $H' \in \mathbb{R}^{n \times F}$ ：

$$H' = GCN(H) = g(A(G)HW^T + b) \quad (2.1)$$

其中  $W \in \mathbb{R}^{F' \times F}$ ， $b \in \mathbb{R}^{F'}$  是模型参数， $g$  为非线性激活函数， $A(G)$  为获取图  $G$  结构信息的  $n \times n$  矩阵，GCN 将  $A(G)$  实例化为与归一化图 Laplacian 密切相关的静态矩阵：

$$A_{GCN(H)} = D^{-1/2}AD^{-1/2} \quad (2.2)$$

其中  $A$  是  $G$  的邻接矩阵， $D = \text{diag}(A1)$  是度矩阵。

引入一个简单而又灵活的图信息传播模型  $f(X, A)$ ，我们可以返回到半监督节点分类问题。我们可以放宽通常在基于图的半监督学习中所作的某些假设，方法是对数据  $X$  和底层图结构的邻接矩阵  $A$  对我们的模型  $f(X, A)$  进行条件调整。在邻接矩阵包含不存在于数据  $X$  中的信息的场景中，例如引用网络中的文档之间的引用链接或知识图中的关系时，通常期望这种设置特别强大。总体模型，一个用于半监督学习的多层 GCN。

我们首先在预处理步骤中计算  $\hat{A} = \tilde{D}^{(-1/2)}\tilde{A}\tilde{D}^{(-1/2)}$ 。然后，我们的传播模型采用了简单的形式：

$$Z = f(X, A) = \text{softmax}(\hat{A}\text{ReLU}(\hat{A}XW^{(0)})W^{(1)}) \quad (2.3)$$

这里， $W^{(0)} \in \mathbb{R}^{C \times H}$  是一个具有  $H$  特征映射的隐藏层的输入到输出的权矩阵。 $W^{(1)} \in \mathbb{R}^{C \times H}$  是一个隐输出权矩阵。Softmax 激活函数，定义为  $\text{softmax}(X_i) = \frac{1}{Z} \exp(X_i)$ ， $Z = \sum_i \exp(X_i)$ ，逐行应用。对于半监督的多类分类，对所有标号样本的交叉熵误差进行了评估：

$$\mathcal{L} = - \sum_{l \in Y_L} \sum_{f=1}^F Y_{lf} \ln Z_{lf}, \quad (2.4)$$

其中  $Y_L$  是一组具有标签的节点索引。利用梯度下降训练神经网络权值  $W^{(0)}$  和  $W^{(1)}$ 。

## 2.2 GHSOM 模型

GHSOM 网络是建立在 SOM 的基础上的一种延伸模型。简单地说，自组织映射是一种实现竞争学习的人工神经网络，可以认为是一种无监督学习形式。在地图上，神经元沿着  $x_{dim}$  和  $y_{dim}$  维数的矩形网格排列。对于每个训练实例  $\vec{x}_k, k = 1, 2, 3, \dots, M$ ， $M$  是训练实例的数目。

1、竞争步骤，在矩形网格上找到特定训练实例的最佳匹配神经元，

$$c = \operatorname{argmin}_i (||\vec{x}_i - \vec{x}_k||), \quad (2.5)$$

其中  $i, k = 1, 2, 3, \dots, N$  是 MAP 神经元上的一个指数， $N = x_{dim} \times y_{dim}$  是网格上的神经元数， $\vec{m}_i$  是  $i$  索引的神经元。最后， $c$  是映射上最佳匹配神经元的指标。

2、更新步骤，训练实例  $\vec{x}_k$  影响最佳匹配神经元  $\vec{m}_c$  及其邻域。更新步骤可以由地图上神经元的以下更新规则表示，

$$\vec{m}_i \leftarrow \vec{m}_i - \eta \vec{\delta}_i h(c, i) \quad (2.6)$$

即  $i = 1, 2, \dots, N$ ，这里  $\vec{\delta} = \vec{m}_i - \vec{x}_k$ ， $\eta$  是学习速率， $h(c, i)$  是一个损失函数，

$$h(c, i) = \begin{cases} 1 & \text{if } i \in \Gamma(c), \\ 0 & \text{otherwise,} \end{cases} \quad (2.7)$$

其中  $\Gamma(c)$  是最佳匹配神经元  $\vec{m}_c$  与  $c \in \Gamma(c)$  的邻域。通常情况下，邻居是时间的函数，它的大小在训练期间衰减。最初，神经元  $\vec{m}_c$  的邻域包括地图上的所有其他神经元，

$$\Gamma(c)|_{t=0} = 1, 2, \dots, N, \quad (2.8)$$

随着训练的进行， $\vec{m}_c$  的邻域缩小到仅限于神经元本身，

$$\Gamma(c)|_{t \gg 0} = c, \quad (2.9)$$

在这里，像以前一样， $N = x_{dim} \times y_{dim}$  是映射上的神经元数。这意味着每个最佳匹配神经元的更新规则最初都有一个很大的影响范围，逐渐缩小到影响域只包含最佳匹配神经元本身的程度。

对每个训练实例重复上述两个训练步骤，直到给定的映射收敛为止。

在 SOM 中，输出层的类神经网络单元是以一维或二维矩阵的方式排列，并根据输入向量彼此竞争，获胜的神经元称之为优胜神经元，其可获得调整权重向量的机会，经过数个迭代的权重调整后，输出层的神经元便会呈现出代表输入向量特征的拓扑结构 (topology)。

增长阶层式自组织映像图 (以下简称 GHSOM) 增长阶层式自组织映像图模型是 SOM 的一种延伸模型，整体呈现多层次的阶层式结构，每层又由数个能自我调适的 SOM 所构成...

GHSOM 克服了固定大小数据的映射方式的限制；因此，即使是在没有先验知识的情况下，依然可以提供最令人满意的结果的网络架构。由于所有数据都映射在不同的二维空间中，因此层次关系清晰。

## 第3章 实验设置

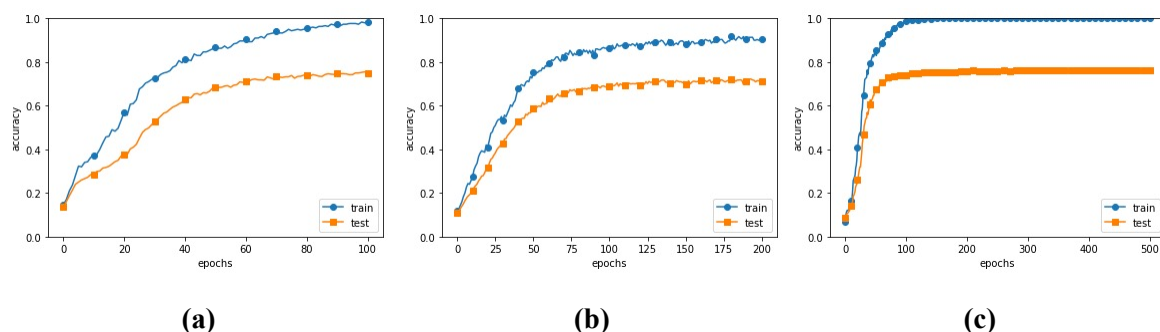
我们利用大规模的真实数据集进行了实验，以定量评估所提出的算法框架。

### 3.1 数据集

我们的实验在微博的社交网络上进行。微博是中国最受欢迎的社交平台之一。数据集来自网络爬虫<sup>[8]</sup>，数据包含了微博 id、微博作者 id、微博发布时间以及评论数转发数等相关多的特征。完整的数据格式和各字段意义见附录。

使用 tkipf-GCN 框架<sup>[7]</sup>还支持多个图实例 (可能不同大小) 的分批分类，每个实例都有一个邻接矩阵。最好是将各自的特征矩阵连接起来，并建立一个 (稀疏) 块对角矩阵，其中每个块对应于一个图实例的邻接矩阵。

这是模型的运行结果图，随着迭代次数的增加，过了 500 个 epoch 左右后，用训练数据测量到的识别精度几乎都为 100%。但是，对于测试数据，离 100% 的识别精度还有较大的差距。如此大的识别精度差距，是只拟合了训练数据的结果。出现了过拟合现象 (如图3.1)，再经过正则化修正之后，选取迭代次数为 200 性能的 GCN 网络。该网络预测的准确率稳定



**图 3.1** 预测准确率,(a) 迭代次数为 100 次的 GCN 预测准确率, (b) 迭代次数为 200 次, (c) 迭代次数为 500 次。

在 75% 左右, 满足我们的实验要求。一份无特征缺失的数据保障了实验的进行。

这里，我们将处理过后的数据进行时间和空间尺度上的分析，下面是分析结果的可视化。图3.2是微博用户的位置的可视化。从图3.2中可以看出，该事件转发评论的微博用户主要集中在沿海地区，江苏尤为集中，内陆以四川、河南为代表，总体来说经济发达地区对此事件关注度普遍较高。因此，我们认为地理距离也是衡量社交相似性的一个重要特征。具体可以查看数据中省份城市编码，对照新浪微博的开发文档中的省份城市编码表。

时间尺度分析：如图3.3所示。图3.3a反映了各个转发层数随着时间的分布。通过观察我们可以发现，大多数意见领袖都处于第一层转发者，事件最多经过了 7 层转发。可见转发层数也是衡量一个事件影响力的重要特征。图3.3b反映了各个转发节点随着时间的分布。

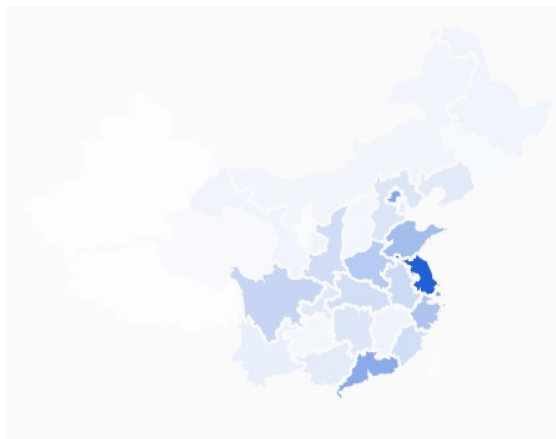


图 3.2 微博用户的地理分布

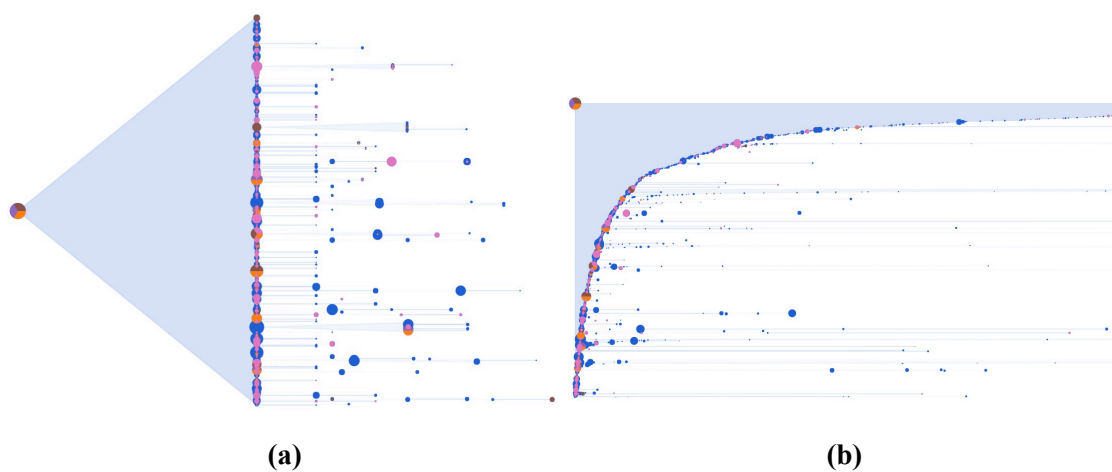


图 3.3 (a) 树形图, (b) 船帆图。

关键节点分析：如图??。图??中可以清楚地观察到事件的核心社交圈以及各个子社交圈的意见领袖及其分布。由此我们通过分析意见领袖的领域分布及其在各自领域的影响力来得出各个子社交圈的影响力。这些意见领袖在各自社交圈的影响力也成为了评估事件影响力的一个重要特征。热度分析：

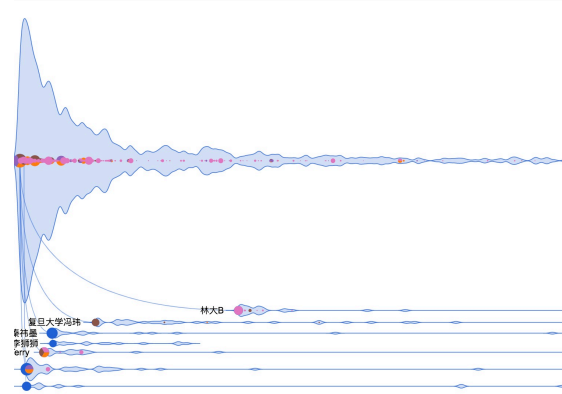


图 3.4 话题热度随时间变化示意图

## 3.2 评估指标

为了定量评估本文的算法，我们使用以下性能指标：

预测性能 我们评估预测性能根据曲线下面积 (AUC)、精度 (Prec)、召回 (Rec) 和 F1 测量 (F1)。

参数灵敏度 我们分析了我们的模型和测试不同的超参数选择如何影响预测性能。

案例研究 我们使用具体案例来进一步证明和解释我们提出的算法的有效性。

## 3.3 算法比较

我们将 GHSOM 与几个算法进行比较。

逻辑回归 (LR) 使用逻辑回归 (LR) 来训练。该模型考虑了三类特征：(1) 用户的顶点特征；(2) 预训练网络嵌入；(3) 手工制作的网络特征。

支持向量机 (SVM) 使用了支持向量模型。以线性核为分类模型的机器 (SVM)。该模型使用与逻辑回归 (LR) 相同的特征。

## 第4章 实验结果

我们比较了上一节中所提到的方法的预测性能，并列出了表4 (AUC)、精度 (Prec)、召回 (Rec) 和 F1 测量 (F1)。并结合两个实际的案例来探索其应用价值。请见表

### 4.1 参数调整

GHSOM 中的超参数主要有初始映像图大小、广度参数和深度参数这三个。通过实际的数据，观察不同的初始映像图大小会在多大程度上影响神经网络的学习。这里基于不同初始映像图大小进行实验。结果如图4.1所示。

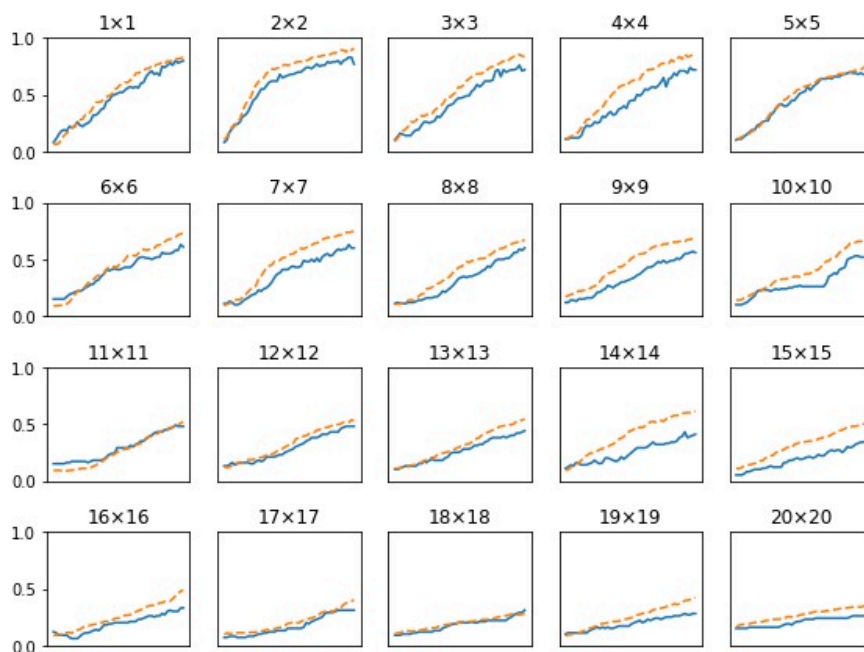


图 4.1 实线是验证数据的识别精度，虚线是训练数据的识别精度

从图4.1中，按识别精度从高到低的顺序排列了验证数据的学习的变化。从图中可知，直到“4×4”左右，学习进行得都很顺利。因此，我们来观察一下“4×4”之前的超参数的值(学习率)，结果如下表4.1所示。从这个结果可以看出，学习率在 0.001 到 0.01 时，学习可以顺利进行。

表 4.1 实验结果

size	val acc	lr
1×1	0.83	0.0092
2×2	0.78	0.00956
3×3	0.77	0.00571
4×4	0.74	0.00626

## 4.2 算法应用

本章节将从舆情监控和广告投放两方面的案例进行分析。

### 4.2.1 舆情监控

在这一部分中，将本文算法得出的结果与微博名下的大数据监控平台微热点的相关分析结果做比较。微热点将微博话题简单的分成了政治、经济、法治、教育、商业、民生、医疗、交通、文娱和体育十大板块。

我们根据系统架构所提出的方法，提取 2019 年 3 月 21 日下午 14 时 48 分左右江苏盐城爆炸事件相关微博，该事件至论文完成时已经累积 2.6 亿次阅读，属于突发事件。部分微博机构、用户的影响力占比如下表所示。

### 4.2.2 广告投放

某著名微博营销网站采用的用户投放报价方式，是基于传统的转发、评论、点赞，如图所示..

## 第 5 章 相关工作

我们的研究与大量关于社会影响分析和图形表征学习的文献密切相关...



## 第 6 章 结论

在这项工作中，研究了社会影响的时空分布、领域分布、地域分布等问题。我们从机器学习的角度来阐述这个问题，并结合最新开发的网络嵌入、图形卷积技术，提出了一个基于图形的分析框架。我在微博上测试了该算法的框架。广泛的实验分析表明，GHSOM 在预测话题社会影响方面明显优于。本文探讨了 GHSOM 网络表征学习在社会影响分析中的潜力，并尝试解释微博话题的动态社会影响...

## 第 7 章 使用简介

### 7.1 数学公式、图表、参考文献等功能

#### 7.1.1 数学公式

比如 Navier-Stokes 方程:

$$\begin{cases} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{V}) = 0 \\ \frac{\partial(\rho \mathbf{V})}{\partial t} + \nabla \cdot (\rho \mathbf{V} \mathbf{V}) = \nabla \cdot \boldsymbol{\sigma} \\ \frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\rho E \mathbf{V}) = \nabla \cdot (k \nabla T) + \nabla \cdot (\boldsymbol{\sigma} \cdot \mathbf{V}) \end{cases} \quad (7.1)$$

数学公式常用命令请见 [WiKibook Mathematics](#)。artracom.sty 中对一些常用数据类型如矢量矩阵等进行了封装，这样的好处是如有一天需要修改矢量的显示形式，只需单独修改 artracom.sty 中的矢量定义即可实现全文档的修改。

#### 7.1.2 表格

请见表 7.1。制表的更多范例，请见 [WiKibook Tables](#)。

表 7.1 这是一个样表。

**Table 7.1** This is a sample table.

Row number	This is a multicolumn							
Row 1	1	2	4	5	6	7	8	
Row 2	1	2	4	5	6	7	8	
Row 3	1	2	4	5	6	7	8	
Row 4	1	2	4	5	6	7	8	

#### 7.1.3 图片插入

插图的参考代码在 `Tex/Commands.tex` 亦进行了汇集。

#### 7.1.4 算法

如见算法 1，详细使用方法请参见文档 [algorithmicx](#)。

#### 7.1.5 参考文献引用

参考文献引用过程以实例进行介绍，假设需要引用名为“Document Preparation System”的文献，步骤如下：

**Algorithm 1** Euclid's algorithm

---

```

1: procedure Euclid( $a, b$ ) ▷ The g.c.d. of  $a$  and  $b$ 
2:    $r \leftarrow a \bmod b$ 
3:   while  $r \neq 0$  do ▷ We have the answer if  $r$  is 0
4:      $a \leftarrow b$ 
5:      $b \leftarrow r$ 
6:      $r \leftarrow a \bmod b$ 
7:   end while
8:   return  $b$  ▷ The gcd is  $b$ 
9: end procedure

```

---

1) 使用 Google Scholar 搜索 Document Preparation System, 在目标条目下点击 Cite, 展开后选择 Import into BibTeX 打开此文章的 BibTeX 索引信息, 将它们 copy 添加到 ref.bib 文件中 (此文件位于 Biblio 文件夹下)。

2) 索引第一行 @article{lamport1986document, 中 lamport1986document 即为此文献的 label (中文文献也必须使用英文 label, 一般遵照: 姓氏拼音 + 年份 + 标题第一字拼音的格式), 想要在论文中索引此文献, 有两种索引类型:

文本类型: `\citet{lamport1986document}`。正如此处所示 Zhao<sup>[9]</sup>;

括号类型: `\citep{lamport1986document}`。正如此处所示<sup>[9]</sup>。

多文献索引用英文逗号隔开:

`\citep{lamport1986document, chen2005zhulu}`。正如此处所示<sup>[8,9]</sup>

如此, 即完成了文献的索引, 请查看下本文档的参考文献一章, 看看是不是就是这么简单呢? 是的, 就是这么简单!

不同文献样式和引用样式可在 Thesis.tex 中对 artratex.sty 调用实现, 如:

- `\usepackage[numbers]{artratex}` % 文本: Jones [1]; 括号: [1]
- `\usepackage[super]{artratex}` % 文本: Jones 上标 [1]; 括号: 上标 [1]
- `\usepackage[authoryear]{artratex}` % 文本: Jones (1995); 括号: (Jones, 1995)
- `\usepackage[alpha]{artratex}` % 文本: 不可用; 括号: [Jon95]

若在上标 (super) 模式下, 希望在特定位置将上标改为嵌入式标, 可使用

文本类型: `\citetns{lamport1986document, chen2005zhulu}`。

正如此处所示??]

括号类型: `\citepns{lamport1986document, chen2005zhulu}`。

正如此处所示[??]

参考文献索引更为详细的信息, 请见 [WiKibook Bibliography](#)。

## 7.2 常见使用问题

1. 模板每次发布前, 都已在 Windows, Linux, MacOS 系统上测试通过。下载模板后, 若编译出现错误, 则请遵从 [位于主页底部的用户指南](#)。
2. 模板文档的编码为 UTF-8 编码。所有文件都必须采用 UTF-8 编码, 否则编译后生成的文档将出现乱码文本。若出现文本编辑器无法打开文档或打开文档乱码的问题, 请检查您使用的编辑器对 UTF-8 编码的支持。如果使用 WinEdt 作为文本编辑器 (不推荐使用), 应在其 Options -> Preferences -> wrapping 选项卡下将两种 Wrapping Modes 中的内容:

TeX;HTML;ANSI;ASCII|DTX...

修改为: TeX;UTF-8|ACP;HTML;ANSI;ASCII|DTX...

同时, 取消 Options -> Preferences -> Unicode 中的 Enable ANSI Format。

3. 推荐选择 xelatex 或 lualatex 编译引擎编译中文文档。编译脚本的默认设定为 xelatex 编译引擎。你也可以选择不使用脚本编译, 如直接使用 T<sub>E</sub>X 文本编辑器编译。注: T<sub>E</sub>X 文本编辑器编译的默认设定为 pdf<sub>l</sub>atex 编译引擎, 若选择 xelatex 或 lualatex 编译引擎, 请进入下拉菜单选择。为正确生成引用链接, 需要进行全编译。

### 4. Texmaker 使用简介

- (a) 使用 Texmaker “打开” Thesis.tex。
- (b) 菜单 “选项 (Options)” -> “设置当前文档为主文档 (Define as Master Document)”
- (c) 菜单 “自定义 (User)” -> “自定义命令 (User Commands)” -> “编辑自定义命令 (Edit User Commands)” -> 左侧选择 “command 1”, 右侧 “菜单项 (Menu Item)” 填入 Auto Build -> 点击下方 “向导 (Wizard)” -> “添加 (Add)” : xelatex + bibtex + xelatex + xelatex + pdf viewer -> 点击 “完成 (OK)”
- (d) 使用 Auto Build 编译带有未生成引用链接的源文件, 可以仅使用 xelatex 编译带有已经正确生成引用链接的源文件。
- (e) 编译完成, “查看 (View)” PDF, 在 pdf 中 “ctrl+click” 可链接到相对应的源文件。

5. 模版的设计可能地考虑了适应性。致谢等所有条目都是通过最为通用的

`\chapter{item name}` and `\section*{item name}`

来显式实现的 (请观察 Backmatter.tex), 从而可以随意添加, 放置, 和修改, 如同一般章节。对于图表目录名称则可在 ucasthesis.cfg 中进行修改。

6. 设置文档样式: 在 artratex.sty 中搜索关键字定位相应命令, 然后修改

- (a) 正文行距: 启用和设置 `\linespread{1.5}`, 默认 1.5 倍行距。

- (b) 参考文献行距：修改 `\setlength{\bibsep}{0.0ex}`
- (c) 目录显示 subsection：修改 `\setcounter{tocdepth}{2}`
- (d) 文档超链接的颜色及其显示：修改 `\hypersetup`
- (e) 页眉页脚设定：frontmatterstyle, mainmatterstyle, 和 backmatterstyle 分别定义前言，主要内容，和附录的页眉页脚样式。通过阅读这一部分的代码，可以轻松的理解和修改以获得自定义的样式。命令的详细解释请参见 `fancyhdr` 的用户文档。同时可参见 `ctex` 宏包用户文档。
- (f) 设置图 2.3 为图 2-3: 设置

```
\renewcommand{\theequation}{\arabic{chapter}-\arabic{equation}}
\renewcommand{\thefigure}{\arabic{chapter}-\arabic{figure}}
\renewcommand{\thetable}{\arabic{chapter}-\arabic{table}}
```

## 7. 字体控制。文档内字体切换方法：

- 宋体：飞扬跋扈 或 飞扬跋扈
- 粗宋体：飞扬跋扈 或 飞扬跋扈
- 黑体：飞扬跋扈 或 飞扬跋扈
- 粗黑体：飞扬跋扈 或 飞扬跋扈
- 仿宋：飞扬跋扈 或 飞扬跋扈
- 楷体：飞扬跋扈 或 飞扬跋扈

由于缺乏一个统一的被各个操作系统所默认携带的完备的中文字体库，`ctex` 针对不同的操作系统而调用各系统上所对应的一类中文字体库。由于很多操作系统的字库往往缺乏原生态的加粗宋体字重，有时会发生加粗宋体被黑体所替换的情形，这对封面的字体造成影响。若需要解决这个问题，可采用调用自定义的一个完备字体库的方案。若需设置字体库，请选择 `xelatex` 或 `lualatex` 编译引擎，并设置需要的字体库。

如用 **Times New Roman** 作为英文和数字字体，在 `artratex.sty` 的

```
\RequirePackage{fontspec}
```

行下添加如下英文字体调用命令：

```
\setmainfont{Times New Roman} （设置英文正文字体）
```

```
\setsansfont{Times New Roman} （设置英文标题字体）
```

Windows 和 Mac OS 自带此字体，Linux 需手动安装或使用 FreeSerif 字体作为替代。部分 `LaTeX` 版本，若 `setmainfont` 与 `setsansfont` 调用同一字体名称，会导致字体冲突，进而导致非加粗黑体失效，影响目录及节标题的显示。解决此问题的办法为 1) 不设

置 `setsansfont` 或 2) 调用一个名字不同的 Times 类字体, 如 `setsansfont{Times}`, 其中 Times 字体为 Mac OS 所带有。长图表标题的英文标题的悬挂缩进是针对 Times 类正文字体校准的, 非 Times 类正文字体将存在轻微位置偏差。

如果需要调用一个**自定义的中文字体库**, 方法为:

- 调用 `ctex` 预定义好的备用字库: 在 `Thesis.tex` 中设置

```
\documentclass[doublesided,fontset=fandol]{Style/ucasthesis}%
```

便可调用 `fandol` 这一字体库。L<sup>A</sup>T<sub>E</sub>X 编译系统一般已携带或是能自动下载安装 `fandol` 字库。若不能, 则请手动下载并安装链接所提供的所有字体即可。本模板使用说明文档就是采用调用 `fandol` 中文字库。

- 手动调用系统带有的中文字库: 首先需要查看系统所带有的中文字库及其名称, 也可选择安装可获得的中文字库。假设系统已安装或带有名为 SC 的中文字库 (此字库为 MacOS 所配备, 具备原生态的加粗宋体), 则可在 `artratex.sty` 的

```
\RequirePackage{fontspec}
```

行下添加如下中文字体调用命令:

```
\setCJKmainfont[BoldFont=Songti SC Bold,ItalicFont=Kaiti SC]{Songti SC Light}%
\setCJKsansfont{Heiti SC}%
```

字库调用的全面解释可参见 `fontspec` (英文字体调用) 和 `xeCJK` (中文字体调用)。因为模版的设定考虑兼顾不同操作系统 (Windows, Linux, Mac OS), 为了模版的健壮性, 上述字体设置和调用方案并未作为原始设定。

8. 封面下划线上的文本不居中下划线, 这是因为那些下划线前面还有字头, 导致文本只能在页面居中和在下划线上居中二选一。当前封面采取页面居中。如需要调整文本在下划线上的位置, 可用 `\hspace{+/- n.0em}` 命令来插入或删除 `n` 个空格, 进行手动调整, 比如

```
\advisor{\hspace{+3.0em} xxx~研究员~xxx单位}
```

这个解决方案是很不优雅, 但问题本质还是样式的设计问题。有时下划线看上去粗细不一致, 这是显示的问题, 打印正常。

9. 对于电子档的论文, 在 `Thesis.tex` 的 `documentclass` 中, 可使用 `singlesided` 来减少空白页。而对于打印版, 可考虑 `printcopy` 选项使奇偶页的排版在打印装订后更美观。
10. 部分所也许对格式有不同设定, 因 L<sup>A</sup>T<sub>E</sub>X 将内容与格式分离, 格式的调可独立于内容进行, 并只需修改模板样式文件完成, 并无风险。

## 附录

## 数据说明

	字段名称	说明
1	<i>id</i>	微博的 ID
2	<i>mid</i>	微博的 MID
3	<i>uid</i>	微博作者的 UID
4	<i>parent</i>	该条微博直接转发微博的 MID
5	<i>type</i>	微博的类型
6	<i>t</i>	微博发布时间, Unix Timestamp
6	<i>comments<sub>count</sub></i>	微博的评论数
7	<i>reposts<sub>count</sub></i>	微博的转发数
8	<i>user<sub>created</sub></i>	微博作者创建时间
9	<i>followers<sub>count</sub></i>	微博作者的粉丝数
10	<i>statuses<sub>count</sub></i>	微博作者的微博数
11	<i>friends<sub>count</sub></i>	微博作者的关注数
12	<i>username</i>	微博作者的用户名
13	<i>user<sub>description</sub></i>	作者的用户描述
14	<i>city</i>	微博作者的城市编号
15	<i>province</i>	微博作者的省份编号
16	<i>gender</i>	作者的性别 m 为男 f 为女 n 为未知
17	<i>verified</i>	微博作者是否已加 V
18	<i>verified<sub>reason</sub></i>	微博作者加 V 原因
19	<i>verified<sub>type</sub></i>	微博加 V 类型
20	<i>original<sub>text</sub></i>	微博文本
21	<i>text</i>	作者本人文本
22	<i>words</i>	分词结果
23	<i>emotion</i>	情感分析

## 伪代码

---

**Algorithm 2** The node2vec algorithm

---

```

1: function LearnFeatures(Graph  $G = (V, E, W)$ , Dimensions  $d$ , Walks per node  $r$ , Walk length
    $l$ , Context size  $k$ , Return  $p$ , In-out  $q$ )
2:    $\pi = \text{PreprocessModifiedWeights}(G, p, q)$ 
3:    $G' = (V, E, \pi)$ 
4:   Initialize  $walks$  to Empty
5:   for  $iter = 1$  to  $r$  do
6:     for all nodes  $u \in V$  do
7:        $walk = \text{node2vecWalk}(G', u, l)$ 
8:        $walk$  to  $walks$ 
9:     end for
10:  end for
11:   $f = \text{StochasticGradientDescent}(k, d, walks)$ 
12:  return  $f$ 
13: end function

1: function node2vecWalk(Graph  $G' = (V, E, \pi)$ , Start node  $u$ , Length  $l$ )
2:   Initialize  $walks$  to  $[u]$ 
3:   for  $walk\_iter = 1$  to  $l$  do
4:      $curr = walk[-1]$ 
5:      $V_{curr} = \text{GetNeighbors}(curr, G')$ 
6:      $s = \text{AliasSample}(V_{curr}, \pi)$ 
7:     Append  $s$  to  $walk$ 
8:   end for
9:   return  $walk$ 
10: end function

```

---



## 参考文献

- [1] BOND R M, FARISS C J, JONES J J, et al. A 61-million-person experiment in social influence and political mobilization[J]. *Nature*, 2012, 489: 295 – EP.
- [2] KEMPE D, KLEINBERG J, TARDOS E. Maximizing the spread of influence through a social network[J]. *Theory of Computing*, 2015, 11(4): 105–147. DOI: [10.4086/toc.2015.v011a004](https://doi.org/10.4086/toc.2015.v011a004).
- [3] MATSUBARA Y, SAKURAI Y, ADITYA PRAKASH B, et al. Rise and fall patterns of information diffusion: Model and implications[J]. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012. DOI: [10.1145/2339530.2339537](https://doi.org/10.1145/2339530.2339537).
- [4] LI C, MA J, GUO X, et al. Deepcas: an end-to-end predictor of information cascades [J/OL]. *CoRR*, 2016, abs/1611.05373. <http://arxiv.org/abs/1611.05373>.
- [5] Yu L, Asur S, Huberman B A. Artificial Inflation: The True Story of Trends in Sina Weibo [J]. *arXiv e-prints*, 2012: arXiv:1202.0327.
- [6] TANG X, MIAO Q, QUAN Y, et al. Predicting individual retweet behavior by user similarity: A multi-task learning approach[J]. *Knowledge-Based Systems*, 2015, 89: 681 – 688. DOI: <https://doi.org/10.1016/j.knosys.2015.09.008>.
- [7] KIPF T N, WELING M. Semi-supervised classification with graph convolutional networks[C]//*International Conference on Learning Representations (ICLR)*. [S.l.: s.n.], 2017.
- [8] REN D, ZHANG X, WANG Z, et al. Weiboevents: A crowd sourcing weibo visual analytic system[C]//*Pacific Visualization Symposium (PacificVis) Notes*, 2014 IEEE. [S.l.: s.n.], 2014: 330–334.
- [9] ZHAO T C S. *Machine learning in complex networks*[M]. Switzerland: Springer Netherlands, 2016.
- [10] Grover A, Leskovec J. node2vec: Scalable Feature Learning for Networks[J]. *arXiv e-prints*, 2016: arXiv:1607.00653.
- [11] QIU J, TANG J, MA H, et al. Deepinf: Social influence prediction with deep learning[C]//*Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD' 18)*. [S.l.: s.n.], 2019.

## 致 谢

值此论文完成之际，谨在此向多年来给予我关心和帮助的老师、学长、同学、朋友和家人表示衷心的感谢！