富信息网络表示学习及典型应用问题研究

（申请清华大学工学博士学位论文）

培 养 单 位：计 算 机 科 学 与 技 术 系
学　　科：计 算 机 科 学 与 技 术
研　究　生：杨　成
指 导 教 师：孙　茂　松　教　授

二〇一九年六月

Network Representation Learning and Typical Application Problems of Information-Rich Networks

Dissertation Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Doctor of Philosophy

in

Computer Science and Technology

by

Yang Cheng

Dissertation Supervisor : Professor Sun Maosong

June, 2019

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容；（3）根据《中华人民共和国学位条例暂行实施办法》，向国家图书馆报送可以公开的学位论文。

本人保证遵守上述规定。

（保密的论文在解密后应遵守此规定）

作者签名：_____　　导师签名：_____

日　　期：_____　　日　　期：_____

摘 要

网络是表达对象与对象间关系的常用数据形式，在人们的日常生活与工作学习中无处不在。除去网络的拓扑结构信息之外，真实的网络数据中一般还包含着根据节点的属性、行为等产生的丰富信息，统称为富信息网络。随着互联网技术和移动智能设备的发展，富信息网络的数据规模飞速增长，并带来了丰富的应用任务和巨大的市场价值。在富信息网络数据的规模及其相关应用的研究需求日益增长的同时，数据驱动的深度学习技术已经在计算机视觉、自然语言处理等多个领域取得了巨大的成功。如何让已经在多个领域展示出其有效性的机器学习，特别是深度学习技术，服务于富信息网络数据及其典型应用已经在近年来成为人工智能领域的研究热点。

传统的邻接矩阵形式的网络表示具有维度过高和数据稀疏两大缺点，使得研究者们无法在网络数据上应用机器学习和深度学习技术。因此，研究者们转而将网络中的节点编码为低维稠密的向量表示，称为网络表示或者网络嵌入。为网络中的节点学习其向量表示的任务称为网络表示学习。本文针对现有的网络表示学习工作的缺点和不足，系统性地进行了以下五个工作：

针对缺乏对于已有网络表示学习算法的理论分析的问题，本文提出了**网络表示学习的统一框架和增强算法**。本工作将大多数现有的只考虑拓扑结构信息的网络表示学习方法总结为一个统一的两步框架：邻近度矩阵构造和降维，并进一步提出了网络嵌入更新（NEU）算法，该算法从理论上隐含地近似了高阶邻近度，可以应用于已有网络表示学习方法以提高它们的性能。

针对现有网络表示学习方法忽略了网络拓扑结构以外的丰富信息的问题，本文提出了**结合富特征信息的网络表示学习**。受前一工作中得到的最先进的网络表示学习算法实际上等同于一种特殊的矩阵分解的结论的启发，该工作以文本特征为例，在矩阵分解的框架下将节点的特征信息结合到网络表示学习中。

针对现有网络表示学习方法难以应用于相对复杂的典型应用问题的缺点，本文以网络表示学习技术作为模型底层，并根据特定的富信息网络场景利用包括循环神经网络、卷积神经网络在内的深度学习模型进行建模，在推荐系统和传播预测两个富信息网络典型应用问题中，创新性地提出了**基于位置的社交网络的推荐系统**、**微观层面的信息传播预测**和**多层面的信息传播预测**的应用模型。

关键词：富信息网络，网络表示学习；网络嵌入；推荐系统；信息传播预测

Abstract

Networks, which are an essential data type to represent objects and their relationships, are widely used in our daily lives and academic researches. Besides the topological structure, a real-world network usually contains rich information generated by the attributes or behaviors of the nodes in the network. These networks are called information-rich networks. With the development of Internet technologies and mobile devices, the scale of information-rich network data has grown rapidly, which leads to many task scenarios and huge market value. Meanwhile, the data-driven deep learning technology has achieved great success in many fields such as computer vision and natural language processing. It has become a research hotspot of artificial intelligence to study how to adopt machine learning (especially deep learning) technology for information-rich networks in recent years.

However, traditional network representation of adjacency matrix has two major disadvantages: high dimensionality and data sparsity. Both disadvantages make it impossible to apply machine learning or deep learning techniques for network data. Therefore, researchers turned to represent nodes in the network by low-dimensional dense vectors, i.e., network representations or network embeddings. The task of learning nodes' vector representations is called network representation learning. To address the problems in previous network representation learning algotirhms, we systematically carry out the following five studies:

For the problem of lacking theoretical analysis of existing network representation learning methods, we propose **a unified framework for network representation learning and enhancement algorithm**. We summarize most of existing network representation learning methods into a unified two-step framework, i.e., proximity matrix construction and dimension reduction. We further propose Network Embedding Update (NEU) algorithm which implicitly approximates high-order proximity in theory and can be applied for any network representation learning method to improve their performances.

For the problem of ignoring other information besides network structure, we propose **network representation learning with rich feature information**. Inspired by the conclusion from the first work, i.e., the state-of-the-art network representation learning algorithm is actually equivalent to matrix factorization, we incorporate text features of

nodes into network representation learning under the framework of matrix factorization.

For the problem of hard to be adopted for more complicated applications, we use network representation learning as the bottom layer of our models, and employ deep learning models including recurrent neural networks (RNNs) and convolutional neural networks (CNNs) for modeling in two typical applications: recommender system and diffusion prediction. In specific, we propose information-rich network application models for **recommendation systems in location-based social networks**, **microscopic information diffusion prediction** and **multi-scale information diffusion prediction**.

Key words: Information-rich Network; Network Representation Learning; Network Embedding; Recommender System; Information Diffusion Prediction

目 录

主要符号对照表

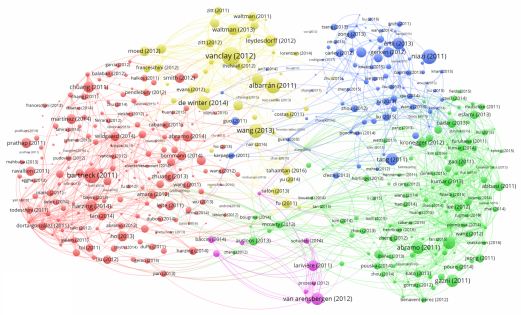
G	网络
V	网络中的节点集合
E	网络中的边集合
A	网络的邻接矩阵
NRL	网络表示学习 (Network Representation Learning)
NE	网络嵌入 (Network Embedding)
RNN	循环神经网络 (Recurrent Neural Network)
GRU	带门循环单元 (Gated Recurrent Unit)
LSTM	长短期记忆 (Long-Short Term Memory)
CNN	卷积神经网络 (Convolutional Neural Network)
RL	强化学习 (Reinforcement Learning)
MF	矩阵分解 (Matrix Factorization)
SVD	奇异值分解 (Singular Value Decomposition)
SVM	支持向量机（Support Vector Machine）
SC	谱聚类（Spectral Clustering）

第 1 章　引言

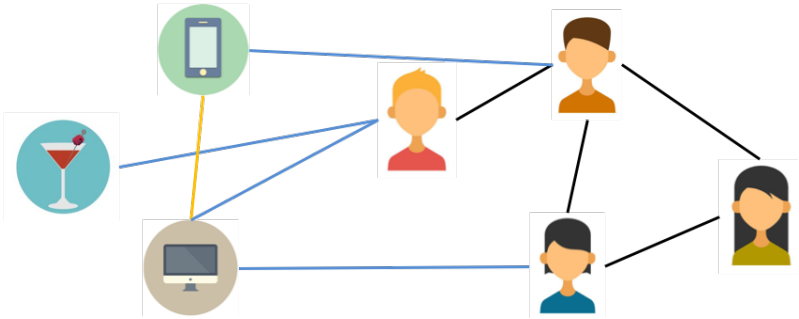
1.1　研究背景

网络是表达对象与对象间关系的常用数据形式，在人们的日常生活与工作学习中无处不在。如图1.1所示，我们平时用社交软件联系好友，用户之间的好友关系构成了社交网络；我们撰写的学术论文被论文数据库收录，论文间的引用关系构成了引文网络；我们在在线购物平台购买商品时，用户和商品之间的浏览、收藏和购买关系等构成了用户和商品之间的异质网络。网络中的节点和节点之间的边组成了网络的拓扑结构。除去网络的拓扑结构信息之外，真实的网络数据中一般还包含着根据节点的属性、行为等产生的丰富信息。例如社交网络中用户对一系列商铺的评价行为产生的用户访问商铺的轨迹信息，引文网络中论文的内容摘要形成的文本信息等等。我们统称这些包含着拓扑结构之外的丰富信息的网络为富信息网络。

(a) 社交网络



(b) 引文网络

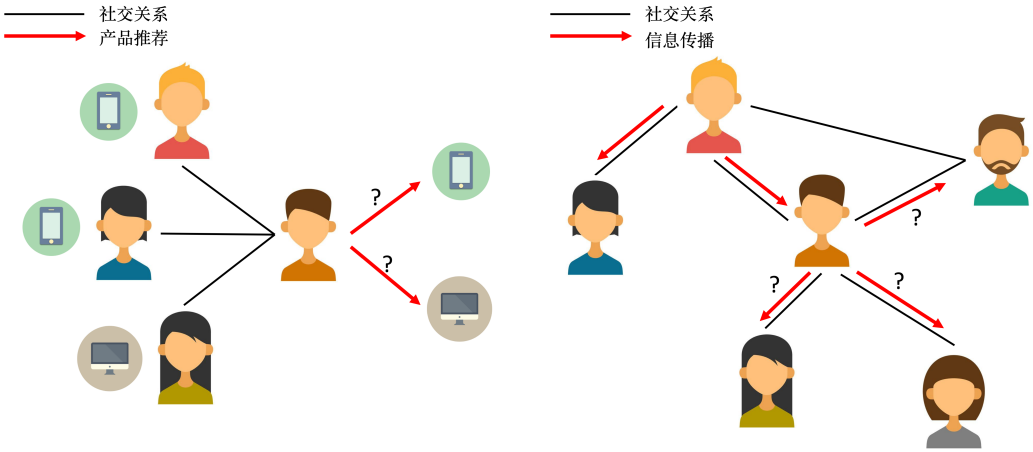


(c) 用户-商品异质网络

图 1.1　无处不在的网络数据场景

近十年中，随着互联网技术和移动智能设备的发展，网络数据的规模也飞速

增长。据统计，截止至 2018 年第四季度，世界最大社交网站 Facebook 月活跃用户数达到 23.2 亿[①]；维基百科页面超链接网络中的总页面数达到 4 千万篇[②]；截止至 2018 年 3 月，在线购物网站淘宝的月活跃用户数达到 6 亿[③]。如此海量的富信息网络数据势必会带来丰富的应用问题和巨大的市场价值。



(a) 推荐系统　　(b) 传播预测

图 1.2　富信息网络的典型应用问题

举例来说，图1.2介绍了富信息网络中的两个典型应用问题:

- 推荐系统：富信息网络数据中的节点经常对应于一个用户。基于用户行为和用户间关系的推荐系统设计是富信息网络场景下的重要应用任务。特别地，随着移动互联网的快速发展，针对移动端场景下的推荐任务也十分值得研究。
- 传播预测：在富信息网络数据中，我们经常可以观察到信息在网络中的传播行为，如社交网络中谣言的传播、用户群体中新产品的推广等等。针对网络中信息的动态传播行为的建模与预测，也是富信息网络的典型应用问题。

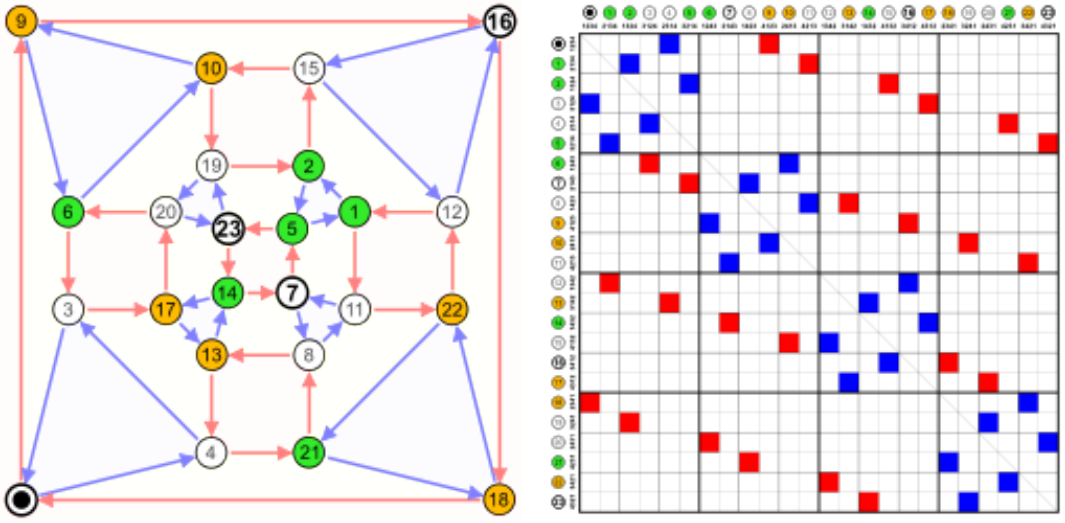
在富信息网络的数据规模及其相关应用的研究需求日益增长的同时，数据驱动的深度学习技术已经在计算机视觉、自然语言处理等多个领域取得了巨大的成功。在 ImageNet 图像识别任务中，以卷积神经网络为基础的深度学习模型分类准确率超过了人类；在机器翻译任务中，以循环神经网络为基础的神经机器翻译模型全面超过了传统的统计机器翻译模型；在围棋 AI 方面，深度学习模型 AlphaGo 打败了人类顶级围棋选手李世石和柯洁。庞大的富信息网络数据规模足以使相关应用任务享受深度学习技术带来的各种优势。如何让机器学习，特别是深度学习

① https://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/
② https://en.wikipedia.org/wiki/Wikipedia:Size_comparisons
③ https://en.wikipedia.org/wiki/Taobao

技术，服务于富信息网络数据及其应用已经在近年来成为人工智能领域的研究热点。

1.2　网络表示学习

不同于图像和文本形式的数据，网络结构的数据具有‘‘全局性’’的特点：网络数据通常由一张大规模的图（graph）构成，无法像图像或者文本数据一样划分成彼此独立的图片或者段落进行处理。因此，网络数据一般用邻接矩阵的形式来表示：邻接矩阵是边长等于网络中节点数的方阵，方阵中第 i 行第 j 列的元素代表了节点 i 和节点 j 之间的链接关系。图1.3展示了邻接矩阵的示例，其中每一行（列）对应一个节点的链接关系，白色格和彩色格分别对应零元素和非零元素。



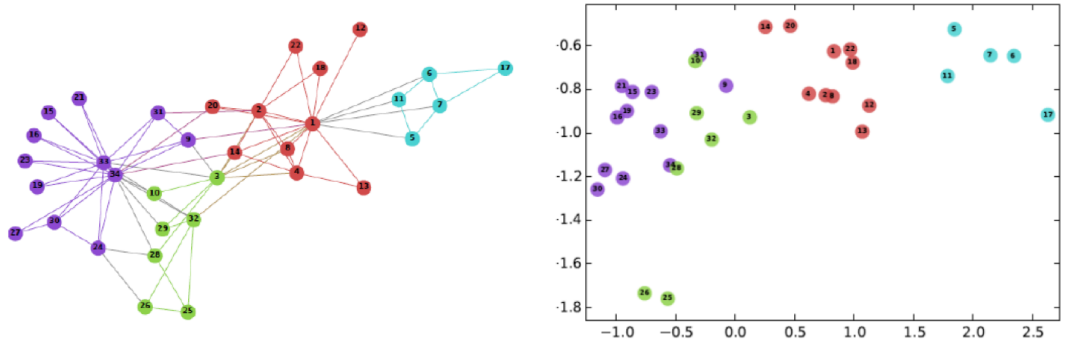
(a) 网络结构　　(b) 邻接矩阵

图 1.3　网络结构及其对应邻接矩阵示例

虽然直观易懂，邻接矩阵形式的网络表示具有维度过高和数据稀疏两大缺点。其中维度过高是指每个节点需要长度等于节点数的向量来表示，加大了后续步骤的计算量；数据稀疏是指矩阵中的非零元素数量稀疏，即编码的信息量少。这两大缺点使得我们无法在传统的邻接矩阵表示上应用机器学习和深度学习技术。

因此，研究者们转而将网络中的节点编码为低维稠密的向量表示，称为网络表示（network representation）或者网络嵌入（network embedding）。网络表示的维度远远小于网络中的节点数。为网络中的节点学习其向量表示的任务称为网络表示学习（network representation learning）。图1.4展示了 DeepWalk[1] 算法进行网络表示学习的示意图：网络表示学习旨在为网络中每个节点学习一个实数向量表示

以编码其拓扑结构，即对于拓扑结构相似的节点，其表示在向量空间中也应该相近。节点的向量表示可看作其特征，送入支持向量机（SVM）等机器学习分类器用于节点分类、链接预测等任务。



(a) 某空手道俱乐部社交关系网络　　(b) DeepWalk 学习的网络表示

图 1.4　某空手道俱乐部社交网络及 DeepWalk[1] 算法学习得到的网络表示可视化

下面我们将简要介绍五个经典的网络表示学习算法：

- **Spectral Clustering**[2] 首先计算归一化的拉普拉斯矩阵，然后计算该矩阵的前 d 个特征向量作为 d 维的节点表示。Spectral Clustering 基于大规模矩阵的特征向量计算，计算复杂度较高，难以应用于大规模数据。
- **DeepWalk**[1] 首先提出使用神经网络模型学习网络表示。DeepWalk 首先从网络中采样随机游走序列，然后将随机游走序列和序列中的节点分别类比为句子和单词，并将最初用于词表示学习的 Skip-Gram[3] 模型用在随机游走序列上来学习网络节点表示，是网络表示学习的代表性算法。
- **Node2vec**[4] 进一步推广了 DeepWalk 算法。Node2vec 在随机游走序列生成时，加入了两个额外的超参数以控制随机游走序列进行局部的宽度优先搜索或者全局的深度优先搜索。Node2vec 以半监督训练的方式调节超参数，取得了比 DeepWalk 更好的效果。
- **LINE**[5] 从网络结构的角度出发，对节点间的一阶邻近度（即直接相连的节点对）、二阶邻近度（通过共同邻居相连的节点对）进行建模来学习大规模网络的节点表示。
- **GraRep**[6] 推广了 k 阶邻近度的概念，并对不同阶的邻近度矩阵用奇异值分解进行降维，最后把每个邻近度矩阵对应的表示拼接起来作为结果。虽然 GraRep 取得了比 DeepWalk 和 LINE 都要好的效果，但是 GraRep 需要进行大规模矩阵的乘法和奇异值分解运算，其计算效率非常低。

上述算法学习得到的节点表示被用作特征，在节点分类和链接预测任务上取得了不错的效果，但仍有很多不足和改进的空间。针对推荐系统、传播预测等典型应用问题的相关工作，我们将在对应的章节中具体展开。

1.3　现有工作存在的问题

现有工作主要存在以下三个方面的问题：

- **缺乏对于已有网络表示学习算法的理论分析**。现有的网络表示学习算法模型多样，如基于特征向量计算的 Spectral Clustering，基于神经网络的 DeepWalk，基于矩阵分解的 GraRep 等。但算法间的比较都是实验层面的效果对比，对于如何进行更好的网络表示学习的理论分析仍是空白。如何提出统一的理论框架对现有网络表示学习算法进行分析、对比和升华，对于相关研究具有非常重要的理论和应用价值。
- **忽略了网络拓扑结构以外的丰富信息**。现有的网络表示学习算法只考虑了网络拓扑结构的信息，而忽略了网络中丰富的其他特征信息，例如社交网络中用户发布的文本图片信息、引文网络中论文的文本信息等等。这些有价值的特征信息应当被纳入网络表示学习算法框架当中。但是如何对网络结构信息和其他特征信息进行有机结合，做到 $1 + 1 > 2$，仍然是一个巨大的挑战。
- **难以应用于相对复杂的典型应用问题**。现有的网络表示学习算法主要基于任务无关的无监督训练，即在学习节点表示的过程中不考虑后续的应用问题，只适用于节点分类和链接预测等相对简单的任务场景。对于推荐系统、传播预测等富信息网络中的典型应用场景，现有的缺乏针对性的算法很难取得很好的效果。如何让网络表示学习技术服务于这些相对复杂的典型应用问题，是具有挑战性的关键研究方向。

1.4　本文的主要研究内容

如图1.5所示，本文针对现有工作中三个方面的问题，从富信息网络表示学习和典型应用问题研究两个层面，系统性地进行了以下五个工作：

在富信息网络表示学习层面，即在不考虑后续应用场景的情况下无监督地学习网络中的节点表示，本文分别解决了**缺乏对于已有网络表示学习算法的理论分析**和**忽略了网络拓扑结构以外的丰富信息**两个问题：

- **网络表示学习的统一框架和增强算法**：现有的网络表示学习方法都聚焦于只考虑网络拓扑结构的一般网络。本工作将大多数网络的表示学习算法总结为一个统一的两步框架：邻近度矩阵构造和降维。本工作专注于邻近度矩阵

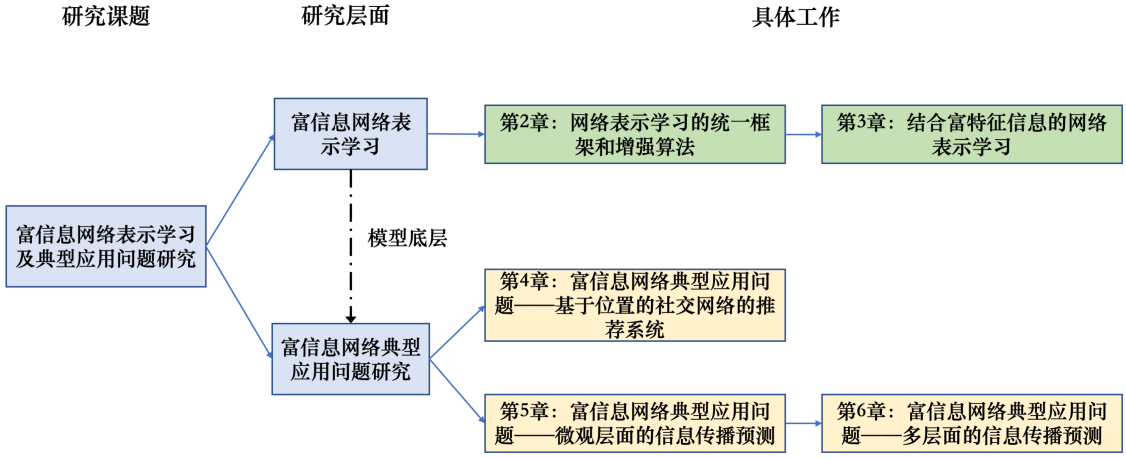


图 1.5　本文研究内容框架

构造步骤的分析，并得出结论，如果在构建邻近度矩阵时探索了更高阶的邻近度，网络表示学习算法可以得到增强。本工作进一步提出了网络嵌入更新（NEU）算法，该算法从理论上隐含地近似了高阶邻近度，并且可以应用于任何网络表示学习方法以提高它们的性能。多标签分类和链接预测任务上的实验结果表明，NEU 在所有三个公开可用的数据集上可以对许多网络表示学习方法进行一致且显著的增强，且运行时间几乎可以忽略不计。

- **结合富特征信息的网络表示学习**：富信息网络中往往还包含着大量的特征信息，其中文本特征是富信息网络中除结构信息以外的常见特征类型。本工作以文本特征为例，提出结合富特征信息进行网络表示学习。根据我们在前一工作中的证明，最先进的网络表示学习算法 DeepWalk[1] 实际上等同于一种特殊的矩阵分解（MF）。受此启发，本工作提出了 TADW 算法，在矩阵分解的框架下将节点的文本特征结合到网络表示学习中。本工作通过将学习得到的节点表示应用于节点多分类任务来评估该方法和各种基线方法。实验结果表明，该方法在所有三个数据集上均优于其他基线，特别是当网络结构噪声较大或者训练数据比例比较小时。

在富信息网络的典型应用问题研究层面，本文以网络表示学习技术作为模型的底层，并根据具体的富信息网络场景利用包括循环神经网络、卷积神经网络在内的深度学习模型进行建模，在推荐系统和传播预测等典型应用任务中解决了现有网络表示学习工作**难以应用于相对复杂的典型应用问题**的缺点：

- **富信息网络典型应用问题——基于位置的社交网络的推荐系统**：面向基于位置的社交网络中的推荐任务，本工作提出了一种可以同时建模社交网络和移动轨迹的神经网络模型。模型由两部分组成：社交网络的构建和移动轨迹的

生成。本工作一方面采用网络表示的方法来构建社交网络，即为每个用户学习网络表示。另一方面建模了四种用户表示来考虑影响移动轨迹生成过程的四个因素，即用户访问偏好，好友关系影响，短期序列影响和长期序列影响。最后，整体模型通过共享的用户网络表示来联合这两个部分。好友推荐和位置推荐任务上的实验结果证明了模型的有效性。特别是在网络结构或轨迹数据稀疏时，该算法对基线的改进更为显著。

- **富信息网络典型应用问题——微观层面信息传播预测**：面向微观层面的信息传播预测任务，本工作提出了神经传播模型（Neural Diffusion Model），简称为 NDM。NDM 首先利用网络表示技术构建模型的底层，并采用深度学习技术，包括注意力机制和卷积神经网络进行级联建模。该模型能够超越之前方法的限制，更好地拟合传播数据，并推广到训练集以外的级联上。四个真实级联数据集上的传播预测任务的实验结果表明，该模型相比于基线方法，F1 值可以有 26% 的相对提升。
- **富信息网络典型应用问题——多层面信息传播预测**：本工作在前一工作的基础上，进一步提出了同时进行宏观和微观层面预测的多层面信息传播预测任务，并提出了一种基于强化学习框架的传播预测模型。强化学习通过解决梯度后向传播中的不可导问题，将宏观传播规模信息引入基于循环神经网络的微观传播模型。为了利用数据中的社交网络结构信息，模型采用了有效的结构上下文提取策略来计算用户的社交网络表示。实验结果表明，本工作提出的模型在三个真实世界数据集的微观和宏观传播预测上都优于最先进的基线模型。

最后，我们对本文工作进行总结，并对未来研究方向作出展望。

第 2 章　网络表示学习的统一框架和增强算法

研究者们近年来已经提出了许多网络表示学习（NRL）方法来学习只考虑拓扑结构的网络中节点的向量表示。在本章[①]中，我们将大多数现有的网络表示学习方法总结为一个统一的两步框架：邻近度矩阵构造和降维。本章工作专注于邻近度矩阵构造步骤的分析，并得出结论，如果在构建邻近度矩阵时探索了更高阶的邻近度，网络表示学习算法可以得到增强。本章工作进一步提出了网络嵌入更新（NEU）算法，该算法从理论上隐含地近似了高阶邻近度，并且可以应用于任何网络表示学习方法以提高它们的性能。多标签分类和链接预测任务上的实验结果表明，NEU 在所有三个公开可用的数据集上可以对许多网络表示学习方法进行一致且显著的增强，且运行时间几乎可以忽略不计。

2.1　问题描述

网络是人们日常生活和学术研究中广泛使用的基本数据类型，例如 Facebook 中的好友网络和 DBLP 中的引文网络[7]。研究者们为开发各种网络应用的机器学习算法付出了巨大努力，如节点分类[8]、标签推荐[9]、异常检测[10] 和链接预测[11]。大多数用于这些应用的监督机器学习算法需要一组信息特征作为输入[4]。人工设计的特征可能适合场景的需要，但却需要大量的人力和专业知识。因此，研究者提出通过最优化来学习特征嵌入的表示学习[12] 以避免特征工程并提高特征的灵活性。就网络分析而言，旨在为网络中的节点学习分布式实值嵌入的网络表示学习（NRL）近年来备受关注[1,4–6]。

本章工作将大多数现有的网络表示学习方法总结为统一的两步框架，包括邻近度矩阵构造和降维。第一步构造了邻近度矩阵 M，其中邻近度矩阵 M 一般表示为归一化邻接矩阵的 K 次多项式形式，M_{ij} 编码了节点 i 和 j 之间的 $k = 1, 2 \ldots K$ 阶邻近度信息。第二步则将邻近度矩阵降维来获得网络嵌入。不同的网络表示学习方法采用了不同的降维算法，例如特征向量计算和 SVD 分解。本章工作对于第一步邻近度矩阵构建的分析表明，当额外的高阶邻近度信息被适当地编码到邻近度矩阵中时，网络嵌入的质量可以得到增强。

然而，高阶邻近度的精确计算是非常耗时的，因此难以应用于大规模的网络数据。因此，我们只能近似高阶邻近度矩阵来学习更好的网络嵌入。为了更高效，

① 本章主要工作以 “Fast Network Embedding Enhancement via High Order Proximity Approximation’’ 为题发表在 2017 年的国际学术会议 “The International Joint Conference on Artificial Intelligence (IJCAI’17)’’ 上。

本章工作使用编码了低阶邻近度信息的网络表示作为基础，以避免重复计算。因此，本章工作提出了网络嵌入更新（NEU）算法，该算法可以应用于任何网络表示学习方法以提高其性能。其背后的思路是，NEU 算法处理后的网络嵌入可以隐含地近似更高阶邻近度信息，且具有理论近似界限，从而获得更好的性能。

为了验证网络嵌入的质量，本章工作在三个公开数据集上进行了多标签分类和链接预测的实验。实验结果表明，在两个评估任务上，现有网络表示学习方法得到的网络嵌入可以在 NEU 增强后得到一致和显著的提升。此外，NEU 的运行时间比常用的网络表示学习算法（如 DeepWalk 和 LINE）训练时间的 1% 还要少，几乎可以忽略不计。

总结来说，本章工作有以下两方面的贡献：

(1) 本章工作将多种现有的网络表示学习方法概括为统一的两步框架，即邻近度矩阵构造和降维，并得出结论，如果适当地将额外的高阶邻近度信息编码到邻近度矩阵中，网络嵌入的质量可以得到增强。

(2) 本章工作提出 NEU 算法来改善任何现有网络表示学习算法产生的网络嵌入的性能。NEU 处理后的网络表示可以间接的近似更高阶的邻近度，且具有理论近似界限。多标签分类和链接预测的实验结果证明了该算法的效率和有效性。

2.2　相关工作

本节将简要介绍现有的 NRL 方法，其中一些方法将在下一节进行详细分析。Spectral Clustering[2] 计算拉普拉斯矩阵的前 d 个特征向量作为 d 维的网络嵌入。DeepWalk[1] 将最初用于词表示学习的 Skip-Gram[3] 模型用在随机游走序列上来学习网络表示。LINE[5] 对节点间的一阶、二阶邻近度进行建模来学习大规模网络的节点表示。GraRep[6] 分解了不同阶的邻近度矩阵，并把每个邻近度矩阵对应的表示拼接起来作为结果。虽然 GraRep 取得了比 DeepWalk 和 LINE 都要好的效果，GraRep 的计算速度却非常低。除了上述关注网络拓扑的网络表示学习方法之外，研究人员还探索了引入标签信息的半监督算法。MMDW[13] 加入不同标签的节点间最大间隔约束来学习半监督网络表示。另一个半监督网络表示学习算法 node2vec[4] 进一步利用生成随机游走时的广度优先搜索和深度优先搜索扩展了 DeepWalk。GCN[14]，DDRW[15] 和 Planetoid[16] 同样用于学习半监督网络表示。SDNE[17] 使用深层神经网络模型学习节点表示。其他扩展包括非对称传递性[18]，社区性质[19] 和异质性[20–24] 的网络嵌入。在本章工作中，我们侧重于只使用网络拓扑结构的最一般情况。

2.3　模型框架

在这一节中，我们首先提出统一的网络表示学习框架，然后证明多个代表性的网络表示学习算法可以归纳为该框架的实例，之后我们根据不同算法在此框架内的横向比较与观察，提出 NEU 算法来增强网络表示的性能。

2.3.1　现有网络表示学习算法的统一框架

在本节中，我们将提出一个统一的算法框架，可以涵盖多个代表性的网络表示学习算法，包括 Spectral Clustering[2]，DeepWalk[1]，LINE[5] 和 GraRep[6]。我们首先解释符号定义并形式化问题，然后介绍 k 阶邻近度的概念。最后，我们提出基于邻近度矩阵分解的网络表示学习框架。

令 $G = (V, E)$ 代表输入网络，其中 V 是节点集合，E 是边集合。网络表示学习旨在为每个节点 $v \in V$ 学习 $r_v \in \mathbb{R}^d$ 的 d 维实数向量表示。不失一般性，在本章中我们假设网络是无权无向图。定义邻接矩阵 $\widetilde{A} \in \mathbb{R}^{|V|\times|V|}$，其中 $\widetilde{A}_{ij} = 1$ 如果 $(v_i, v_j) \in E$，否则 $\widetilde{A}_{ij} = 0$。对角矩阵 $D \in \mathbb{R}^{|V|\times|V|}$，其中 $D_{ii} = d_i$ 表示节点 v_i 的度数。$A = D^{-1}\widetilde{A}$ 是行归一化的邻接矩阵，其每行的和都等于 1。类似的，我们有拉普拉斯矩阵 $\widetilde{L} = D - \widetilde{A}$ 和归一化的拉普拉斯矩阵 $L = D^{-\frac{1}{2}}\widetilde{L}D^{-\frac{1}{2}}$。

2.3.1.1　K 阶邻近度

（归一化的）邻接矩阵和拉普拉斯矩阵刻画了节点间的直接连接关系，即一阶邻近度。注意一阶邻近度矩阵的每个非对角非零元都对应着网络中的一条边。然而真实世界中的网络数据一般是稀疏的，也就是说 $O(E) = O(V)$。因此，一阶邻近度矩阵十分稀疏，不足以对节点间的邻近度关系进行充分的建模。所以，研究者们也探索了更高阶的邻近度关系[1,5,6]。举例来说，二阶邻近度可以用节点间的共同邻居数来刻画。从另一个角度看，v_i 和 v_j 间的二阶邻近度也可以用一个从 v_i 开始的 2 步的随机游走到达 v_j 的概率来建模。直觉上说，如果 v_i 和 v_j 有很多共同邻居，这个概率也会变大。在基于随机游走的概率设置中，我们可以将其推广到 k 阶邻近度[6]：一个从 v_i 开始的 k 步的随机游走到达 v_j 的概率。注意归一化的邻接矩阵 A 是随机游走的单步概率转移矩阵。因此我们可以计算 k 步概率转移矩阵作为 k 阶邻近度矩阵

$$A^k = \underbrace{A \cdot A \ldots A}_{k}, \tag{2-1}$$

其中 A^k_{ij} 是节点 v_i 和 v_j 之间的 k 阶邻近度。

2.3.1.2　网络表示学习框架

在本小节中，我们将具体介绍基于邻近度矩阵降维的两步的网络表示学习框架：

步骤 1: 邻近度矩阵构建。计算编码了 k 阶邻近度矩阵信息的矩阵 $M \in \mathbb{R}^{|V|\times|V|}$，其中 $k = 1, 2 \ldots, K$。例如，$M = \frac{1}{K}A + \frac{1}{K}A^2 \cdots + \frac{1}{K}A^K$ 表示 $k = 1, 2 \ldots, K$ 阶邻近度矩阵的平均组合。邻近度矩阵 M 一般表示为归一化邻接矩阵 A 的 K 次多项式形式。我们将该多项式记为 $f(A) \in \mathbb{R}^{|V|\times|V|}$。这里多项式 $f(A)$ 的幂次 K 对应着邻近度矩阵中编码的最高阶邻近度。注意邻近度矩阵 M 的存储和计算并不一定需要 $O(|V|^2)$ 的复杂度，因为我们只需要计算和存储非零元素。

步骤 2：降维。计算网络嵌入矩阵 $R \in \mathbb{R}^{|V|\times d}$ 和上下文嵌入矩阵 $C \in \mathbb{R}^{|V|\times d}$ 使得它们的乘积 $R \cdot C^T$ 可以近似邻近度矩阵 M。这里不同的算法可能会利用不同的距离函数来最小化 M 和 $R \cdot C^T$ 的差异。例如，我们可以用矩阵 $M - R \cdot C^T$ 的范数来衡量距离并将其最小化。

2.3.2　框架下的算法实例

本小节中，我们将证明多种已有的网络表示学习算法都可以归入此框架。

2.3.2.1　Spectral Clustering

Spectral Clustering[2] 计算归一化拉普拉斯矩阵 L 的前 d 个特征向量作为 d 维的节点表示。特征向量中编码的信息来自于一阶邻近度矩阵 L。注意实对称矩阵 L 可以通过特征分解被分解为 $L = Q\Lambda Q^{-1}$，其中 $\Lambda \in \mathbb{R}^{|V|\times|V|}$ 是对角阵，$\Lambda_{11} \geq \Lambda_{22} \geq \ldots \Lambda_{|V||V|}$ 是特征值，$Q \in \mathbb{R}^{|V|\times|V|}$ 是特征向量矩阵。

我们可以通过设置邻近度矩阵 M 为一阶邻近度矩阵 L，网络嵌入 R 为特征向量矩阵 Q 的前 d 列，上下文嵌入 C^T 为 ΛQ^{-1} 的前 d 行，等价地将 Spectral Clustering 归纳进我们的学习框架。

2.3.2.2　DeepWalk

DeepWalk[1] 将一个广泛使用的词表示方法 Skip-Gram[3]，第一次引入到利用网络结构学习节点表示的研究当中。

DeepWalk 首先生成曾被用于相似度测量[25] 的短随机游走序列。给定随机游走生成的节点序列 $S = \{v_1, v_2, \ldots, v_{|S|}\}$，我们将节点 $v \in \{v_{i-w}, \ldots, v_{i+w}\} \setminus \{v_i\}$ 视为中心节点 v_i 的上下文节点，其中 w 是窗口大小。遵循 Skip-Gram 的想法，DeepWalk

旨在最大化随机游走节点序列 S 中所有节点-上下文对的平均对数概率：

$$\frac{1}{|S|}\sum_{i=1}^{|S|}\sum_{-w\le j\le w, j\neq 0}\log p(v_{i+j}|v_i), \tag{2-2}$$

其中 $p(v_j|v_i)$ 由 softmax 函数定义：

$$p(v_j|v_i)=\frac{\exp(c_{v_j}^T r_{v_i})}{\sum_{v\in V}\exp(c_v^T r_{v_i})}. \tag{2-3}$$

这里 r_{v_i} 和 c_{v_j} 是中心节点 v_i 和它的上下文节点 v_j 的表示向量。换句话说，每个节点 v 有两个表示向量：当 v 作为中心节点时的 r_v 和作为上下文节点时的 c_v 。

之后，DeepWalk 使用 Skip-Gram 和 Hierarchical Softmax 来学习随机游走生成的序列中节点的表示。注意 Hierarchical Softmax[26] 是为了加速 softmax 的变体。

假设 DW 是从随机游走序列中生成的节点-上下文集合，其中 DW 中每个元素是一个节点-上下文对 (v,c)。V 是节点集合，V_C 是上下文节点集合。一般来说，$V=V_C$。

DeepWalk 将节点 v 表示为一个 d 维向量 $r_v\in\mathbb{R}^d$，同时每个上下文节点 $v\in V_C$ 也被表示成 d 维向量 $c_v\in\mathbb{R}^d$。R 为 $|V|\times d$ 的矩阵，其中第 i 行是向量 r_{v_i}；C 为 $|V_C|\times d$ 矩阵，其中第 j 行是向量 c_{v_j}。我们的目标是推导出 M 的解析形式，其中 $M=R\cdot C^T$。

让我们考虑一个节点-上下文对 (v,c)。$N(v,c)$ 表示 (v,c) 在集合 DW 中出现的次数。$N(v)=\sum_{c'\in V_C}N(v,c')$，$N(c)=\sum_{v'\in V}N(v',c)$ 分别代表 v 和 c 在 DW 中出现的次数。

负采样通过随机选取少量负样例，近似了 softmax 函数的概率。前人工作[27] 证明 Skip-Gram 加负采样方法在维度 d 足够大的情况下，等价于间接地分解了词-上下文矩阵 M。M 中每个元素是

$$M_{ij}=\log\frac{N(v_i,c_j)\cdot|DW|}{N(v_i)\cdot N(c_j)}-\log n, \tag{2-4}$$

其中 n 是每个词-上下文对采样的负例数。M_{ij} 可以被理解为平移了 $\log n$ 的词-上下文对 (v_i,c_j) 的点对互信息（PMI）。类似地，我们证明了在更一般的情况下，Skip-Gram 加 softmax 等价于分解了矩阵 M，其中

$$M_{ij}=\log\frac{N(v_i,c_j)}{N(v_i)}. \tag{2-5}$$

我们现在讨论在 DeepWalk 中 M_{ij} 代表了什么含义。显然，采样节点-上下文对的方法会影响矩阵 M。假设网络是无向连通图，窗口大小为 w。我们将基于一个理想的采样方法来讨论 $N(v)/|DW|$，$N(c)/|DW|$ 和 $N(v,c)/N(v)$：首先我们生成一个足够长的随机游走 RW，其中 RW_i 表示 RW 上的第 i 个节点。然后我们将节点-上下文对 (RW_i, RW_j) 加入到 DW 当且仅当 $0 < |i-j| \leq w$。

对于无向图，节点 i 的每次出现在 DW 中会被记录 $2w$ 次。因此，$N(v_i)/|DW|$ 是节点 v_i 在随机游走序列中的出现频率，正好是 v_i 的 PageRank 值。另外，$2wN(v_i,v_j)/N(v_i)$ 是节点 v_j 在节点 v_i 左 w 和右 w 个邻居中出现的期望次数。基于此理解，我们进一步推理 $N(v_i,v_j)/N(v_i)$ 的意义。

注意 PageRank 中的转移矩阵即为行归一化的邻接矩阵 A。我们用 e_i 表示第 i 维是 1，其它维是 0 的 $|V|$ 维的行向量。

假设我们从节点 i 开始随机游走并用 e_i 表示初始状态。那么 e_iA 是随机游走的第一步在全部节点上的概率分布，其中 e_iA 的第 j 个元素是节点 i 走到 j 的概率。因此，e_iA^w 的第 j 个元素是节点 i 用刚好 w 步走到 j 的概率，其中 A^w 是矩阵 A 的 w 次连乘积。所以 $[e_i(A+A^2+\cdots+A^w)]_j$ 是 v_j 出现在 v_i 的右 w 个邻居内的期望次数。我们有

$$\frac{N(v_i,v_j)}{N(v_i)} = \frac{[e_i(A+A^2+\cdots+A^w)]_j}{w}. \tag{2-6}$$

这个等式对于有向图同样成立。因此，$M_{ij} = \log N(v_i,v_j)/N(v_i)$ 是节点 i 在 w 步内随机游走到 j 的平均概率的对数。

因此，DeepWalk 等价于把 $M \in \mathbb{R}^{|V|\times|V|}$ 分解为网络嵌入和上下文嵌入表示的乘积 $R \cdot C^T$，其中

$$M = \log \frac{A+A^2+\cdots+A^w}{w}, \tag{2-7}$$

w 是 Skip-Gram 模型的窗口大小。矩阵 M 刻画了一阶、二阶 $\ldots w$ 阶邻近度的平均。DeepWalk 算法通过基于随机游走生成的蒙特卡罗采样来近似高阶邻近度，而没有直接计算 k 阶邻近度矩阵。

为了将 DeepWalk 融入我们的框架，我们可以设置 $M = f(A) = \frac{A+A^2+\cdots+A^w}{w}$。注意我们忽略了公式 (2-7) 中的对数操作。

2.3.2.3　GraRep

GraRep[6] 对 $k = 1, 2 \ldots, K$ 精确地计算了 k 阶邻近度矩阵 A^k，为每个 k 计算了单独的表示，并将这些表示拼接起来。特别地，GraRep 用 SVD 分解对 k 阶邻近度矩阵 A^k 降维得到 k 阶表示。我们假设 k 阶邻近度矩阵 A^k 被分解为 $U\Sigma S$ 的乘积，其中 $\Sigma \in \mathbb{R}^{|V|\times|V|}$ 是对角矩阵，$\Sigma_{11} \geq \Sigma_{22} \geq \ldots \Sigma_{|V||V|} \geq 0$ 是奇异值，$U, S \in \mathbb{R}^{|V|\times|V|}$ 是正交矩阵。GraRep 定义 k 阶网络嵌入和上下文嵌入 $R_{\{k\}}, C_{\{k\}} \in \mathbb{R}^{|V|\times d}$ 分别为 $U\Sigma^{\frac{1}{2}}$ 和 $S^T\Sigma^{\frac{1}{2}}$ 的前 d 列。因此，k 阶表示的计算也符合我们的框架。但是 GraRep 无法有效的适用于大规模网络[4]：虽然一阶邻近度矩阵 A 是稀疏的，但是直接计算 A^k $(k \geq 2)$ 的开销是 $O(|V|^2)$ 级，无法适应大规模网络数据的需要。

类似地，LINE[5] 也可以被放入此框架。

2.3.3　观察和问题定义

到目前为止，我们已经证明四种代表性的网络表示学习算法可以归纳进我们的两步框架，即邻近度矩阵构造和降维。在本工作中，我们将研究重点放在第一步，并研究如何为网络表示学习定义更好的邻近度矩阵。对于不同降维方法的研究，如 SVD 分解等，将作为未来的工作。

表 2.1　三种网络表示学习方法间的比较。

	SC	DeepWalk	GraRep
邻近度矩阵	L	$\sum_{k=1}^{K} \frac{A^k}{K}$	$A^k, k = 1 \ldots K$
计算过程	精确	近似	精确
效率	快	快	慢
效果	低	中	高

我们在表2.1中总结了 Spectral Clustering (SC)，DeepWalk 和 GraRep 方法间的比较并有以下观察结论。

观察 1：建模更高阶和精确的邻近度矩阵可以提高网络表示的质量。换句话说，如果我们合适地使用更高幂次的多项式邻近度矩阵 $f(A)$，网络表示学习的质量会更好。

从网络表示学习的发展过程中，我们可以看到 DeepWalk 优于 Spectral Clustering，因为 DeepWalk 考虑了更高阶的邻近度矩阵而高阶的邻近度矩阵可以为低阶的邻近度矩阵补充更多的信息。GraRep 优于 DeepWalk 因为 GraRep 精确地计算了 k 阶邻近度矩阵而不是像 DeepWalk 一样使用蒙特卡洛近似。

观察 2：对于大规模网络来说，高阶邻近度矩阵的精确计算并不合适。

GraRep 的主要缺点是精确计算 k 阶邻近度矩阵的复杂度。实际上，高阶邻近度矩阵的计算需要 $O(|V|^2)$ 时间，SVD 分解的复杂度也随着 k 阶邻近度矩阵变的稠密而增长。总体来说，$O(|V|^2)$ 级别的复杂度对于大规模网络来说还是过高了。

第一个观察引导我们在网络表示学习算法中探索更高阶的邻近度矩阵，但第二个观察阻止我们对高阶邻近度矩阵的精确计算。因此，我们转而研究如何有效地从近似的高阶邻近度矩阵中学习网络嵌入的问题。为了更加高效，我们的目标是使用编码了低阶邻近度矩阵信息的网络表示作为基础以避免重复计算。下面我们将该问题形式化。

问题定义：假设归一化邻接矩阵 A 为一阶邻近度矩阵，网络嵌入 R 和上下文嵌入 C，其中 $R, C \in \mathbb{R}^{|V|\times d}$。假设 R 和 C 是由上述网络表示学习框架得到，也就是乘积 $R \cdot C^T$ 近似了 K 次的多项式矩阵 $f(A)$。我们的目标是学习更好的表示 R' 和 C' 使它们的乘积近似了比 $f(A)$ 幂次更高的矩阵 $g(A)$。此外，算法应该在 $|V|$ 的线性时间内完成。注意时间复杂度的下界是 $O(|V|d)$，即嵌入矩阵 R 的参数规模。

2.3.4　近似算法

在本小节中，我们提出了一种简单、快速和有效的迭代更新算法来解决上述问题。

算法：给定超参数 $\lambda \in (0, \frac{1}{2}]$，归一化邻接矩阵 A，我们按下式更新网络表示 R 和上下文表示 C：

$$\begin{aligned} R' &= R + \lambda A \cdot R, \\ C' &= C + \lambda A^T \cdot C. \end{aligned} \tag{2-8}$$

计算 $A \cdot R$ 和 $A^T \cdot C$ 的时间复杂度都是 $O(|V|d)$ 因为矩阵 A 是稀疏的且拥有 $O(|V|)$ 的非零元素数。因此公式 (2-8) 的每轮迭代时间为 $O(|V|d)$。

注意初始表示 R 和 C 的乘积近似了 K 次的多项式邻近度矩阵 $f(A)$。现在我们证明该算法可以得到更好的表示 R' 和 C'，其乘积 $R' \cdot C'^T$ 近似了 $K+2$ 次的多项式邻近度矩阵 $g(A)$，且有矩阵无穷范数定义的近似上界。

定理：给定网络和嵌入表示 R 和 C，我们假设 $R \cdot C^T$ 和邻近度矩阵 $M = f(A)$ 间的近似是有界的 $r = ||f(A) - R \cdot C^T||_\infty$，这里 $f(\cdot)$ 是 K 次的多项式。那么公式 (2-8) 更新后的表示 R' 和 C' 的乘积近似了 $K+2$ 次的多项式 $g(A) = f(A) + 2\lambda A f(A) + \lambda^2 A^2 f(A)$，且近似上界 $r' = (1 + 2\lambda + \lambda^2)r \leq \frac{9}{4}r$。

证明：假设 $S = f(A) - RC^T$，且 $r = ||S||_\infty$。

$$\begin{aligned}
&||g(A) - R'C'^T||_\infty = ||g(A) - (R + \lambda AR)(C^T + \lambda C^T A)||_\infty \\
&= ||g(A) - RC^T - \lambda ARC^T - \lambda RC^T A - \lambda^2 ARC^T A||_\infty \\
&= ||S + \lambda AS + \lambda SA + \lambda^2 ASA||_\infty \\
&\leq ||S||_\infty + \lambda||A||_\infty||S||_\infty + \lambda||S||_\infty||A||_\infty + \lambda^2||S||_\infty||A||_\infty^2 \\
&= r + 2\lambda r + \lambda^2 r.
\end{aligned} \tag{2-9}$$

这里倒数第二个等式根据定义替换了 $g(A)$ 和 $f(A) - RC^T$，最后一个等式是因为 $||A||_\infty = \max_i \sum_j |A_{ij}| = 1$。

在我们的实验设置中，我们假设低阶邻近度的权重应该大于高阶邻近度，因为它们与原始网络更直接相关。因此，给定 $g(A) = f(A) + 2\lambda A f(A) + \lambda^2 A^2 f(A)$，我们有 $1 \geq 2\lambda \geq \lambda^2 > 0$，即 $\lambda \in (0, \frac{1}{2}]$。这个证明表示更新后的节点表示可以在 $\frac{9}{4}$ 倍矩阵无穷范数内，近似幂次更高 2 阶的 $g(A)$。证毕。

算法公式 (2-8) 的更新可以在两个方向上被进一步推广。首先我们可以根据公式 (2-10) 来更新 R 和 C：

$$\begin{aligned}
R' &= R + \lambda_1 A \cdot R + \lambda_2 A \cdot (A \cdot R), \\
C' &= C + \lambda_1 A^T \cdot C + \lambda_2 A^T \cdot (A^T \cdot C).
\end{aligned} \tag{2-10}$$

时间复杂度仍然是 $O(|V|d)$ 但公式 (2-10) 在一次迭代中可以比公式 (2-8) 近似更高阶的邻近度矩阵。我们也可以使用比公式 (2-10) 更复杂的更新，但在我们的实验中，我们使用公式 (2-10) 作为性价比更高的选择。

另一个方向是，更新公式可以被执行 T 轮来获得更好的结果。但是，近似上界会随着 T 呈指数增长，所以该更新无法被执行无限次。注意 R 和 C 的更新是完全独立的。因此对于网络表示学习我们只需要更新表示 R。我们将算法命名为 Network Embedding Update (NEU)。NEU 避免了高阶邻近度矩阵的精确计算，但可以产生实际上近似了高阶邻近度的网络嵌入。因此，我们的算法可以有效地提高网络嵌入的质量。直观上，公式 (2-8) 和 (2-10) 让学习得到的表示进一步传播给每个节点的邻居。所以可以编码更远距离间的节点间邻近度。

2.4　实验结果

我们在两个任务上评估了网络嵌入的质量：多标签分类和链接预测。我们对基线方法学习的嵌入执行网络嵌入更新算法（NEU），并报告测试性能和运行时间。

2.4.1　数据集

我们在三个公开数据集上进行了实验：Cora[8]，BlogCatalog 和 Flickr[2]。我们假设全部三个数据都是无向无权图。

Cora 包含来自 7 个分类的 2,708 篇机器学习论文和 5,429 条它们之间的链接。这些链接是文档之间的引用关系。每篇文章有且仅有一个类别标签。每篇文章拥有 1,433 维的二进制文本特征向量，表示对应单词是否出现在文章之中。

BlogCatalog 包含 10,312 个博主以及 333,983 条博主间好友关系。标签是博主的话题兴趣。数据集中有 39 个标签且一个博主可能有多个标签。

Flickr 包含 80,513 名照片分享网站的用户和 5,899,882 条用户间好友关系。标签是用户和兴趣组间的从属关系。数据集中有 195 个标签且每个用户可能有多个标签。

2.4.2　基线方法和实验设置

我们考虑了许多基线来证明 NEU 算法的有效性和鲁棒性。对于所有方法和数据，我们设置表示维度 $d = 128$。

Graph Factorization (GF) 直接用 SVD 分解了归一化邻接矩阵 A 来获得网络嵌入。

Spectral Clustering (SC)[2] 计算归一化拉普拉斯矩阵的前 d 个特征向量作为 d 维表示。

DeepWalk[1] 生成随机游走序列并使用 Skip-Gram 模型学习表示。DeepWalk 除维度 d 外有三个超参数：窗口大小 w，随机游走长度 t 和每个节点游走数 γ。随着这些超参数的增加，训练样本的数量和运行时间将增加。我们评估了 DeepWalk 的三组超参数，原作者所实现代码的默认参数 DeepWalk$_{low}$：$w = 5, t = 40, \gamma = 10$，node2vec[4] 中的设置 DeepWalk$_{mid}$：$w = 10, t = 80, \gamma = 10$ 以及原始论文中的设置[1]DeepWalk$_{high}$：$w = 10, t = 40, \gamma = 80$。

LINE[5] 是一种可扩展的网络表示学习算法，它使用两个独立的网络表示来建模节点之间的一阶和二阶邻近度 LINE$_{1st}$ 和 LINE$_{2nd}$。我们使用默认的超参数设定，除了总训练样例数 $s = 10^4|V|$，使得 LINE 和 DeepWalk$_{mid}$ 有相当的运行时间。

TADW[28] 在矩阵分解的框架下将文本信息引入 DeepWalk。我们在 Cora 数据集上加入此基线方法。

node2vec[4] 用随机游走的宽度优先和深度优先搜索推广了 DeepWalk，是一个半监督的网络表示学习算法。我们使用其论文中同样的超参数 $w = 10, t = 80, \gamma = 10$，并对其他两个超参数 $p, q \in \{0.25, 0.5, 1, 2, 4\}$ 进行网格搜索作为半监督训练。

GraRep[6] 对 $k = 1, 2 \ldots, K$ 精确地计算了 k 阶邻近度矩阵 A^k，为每个 k 计算了单独的表示，并将这些表示拼接起来。因为 GraRep 计算效率低[4]，我们只对最小的数据集 Cora 测试了 GraRep[6]。我们设置 $K = 5$，因此 GraRep 有 $128 \times 5 = 640$ 维。

实验设置：我们如下设置 NEU 的超参数：对于所有三个数据集 $\lambda_1 = 0.5, \lambda_2 = 0.25$，对于 Cora 和 BlogCatalog，$T = 3$；对于 Flickr，$T = 1$。这里 λ_1, λ_2 是出于低阶邻近度矩阵应该有更高权重的假设的经验设置，T 是 10% 随机验证集上效果开始下降的最大轮数。实际上，如果我们对于下游任务没有任何先验知识，我们可以直接设置 $T = 1$。实验在单个 CPU 上执行以便于运行时间比较，CPU 类型为 Intel Xeon E5-2620 @ 2.0GHz。

2.4.3　多标签分类

表 2.2　Cora 数据集分类结果

	% 分类准确率			运行时间 (秒)
% 训练比例	10%	50%	90%	
GF	50.8 (**68.0**)	61.8 (**77.0**)	64.8 (**77.2**)	4 (+0.1)
SC	55.9 (**68.7**)	70.8 (**79.2**)	72.7 (**80.0**)	1 (+0.1)
DeepWalk$_{low}$	71.3 (76.2)	76.9 (81.6)	78.7 (81.9)	31 (+0.1)
DeepWalk$_{mid}$	68.9 (**76.7**)	76.3 (82.0)	78.8 (84.3)	69 (+0.1)
DeepWalk$_{high}$	68.4 (**76.1**)	74.7 (80.5)	75.4 (81.6)	223 (+0.1)
LINE$_{1st}$	64.8 (70.1)	76.1 (80.9)	78.9 (82.2)	62 (+0.1)
LINE$_{2nd}$	63.3 (**73.3**)	73.4 (80.1)	75.6 (80.3)	67 (+0.1)
node2vec	76.9 (77.5)	81.0 (81.6)	81.4 (81.9)	56 (+0.1)
TADW	78.1 (84.4)	83.1 (86.6)	82.4 (87.7)	2 (+0.1)
GraRep	70.8 (76.9)	78.9 (82.8)	81.8 (84.0)	67 (+0.3)

对于多标签分类任务，我们随机选择一部分节点作为训练集，剩下的作为测试集。和前人工作[2,29] 一样，我们将网络嵌入视为节点特征，并将它们提供给 LibLinear[30] 实现的一对多 SVM 分类器。我们重复 10 次实验并报告 Macro-F1 和 Micro-F1 的平均分数。因为 Cora 数据集的一个节点只有一个标签，我们只报告分

表 2.3　BlogCatalog 数据集分类结果

	% Macro-F1			运行时间 (秒)
% 训练比例	1%	5%	9%	
GF	6.6 (**7.9**)	9.8 (**11.3**)	10.3 (**12.2**)	19 (+1)
SC	8.4 (**9.3**)	13.1 (**14.8**)	14.5 (**17.0**)	10 (+1)
DeepWalk$_{low}$	11.3 (12.4)	15.9 (17.4)	17.1 (18.6)	100 (+1)
DeepWalk$_{mid}$	11.2 (**13.3**)	16.9 (**19.2**)	18.4 (**20.8**)	225 (+1)
DeepWalk$_{high}$	12.4 (13.6)	18.3 (20.1)	20.4 (22.0)	935 (+1)
LINE$_{1st}$	11.1 (12.2)	16.6 (18.3)	18.6 (20.1)	241 (+1)
LINE$_{2nd}$	10.3 (11.2)	15.0 (16.8)	16.5 (**18.3**)	244 (+1)
node2vec	12.5 (13.0)	19.2 (19.8)	21.9 (22.5)	454 (+1)
	% Micro-F1			运行时间 (秒)
% 训练比例	1%	5%	9%	
GF	17.0 (**19.6**)	22.2 (**25.0**)	23.7 (**26.7**)	19 (+1)
SC	19.4 (20.3)	26.9 (28.1)	29.0 (31.0)	10 (+1)
DeepWalk$_{low}$	24.5 (26.4)	31.0 (33.4)	32.8 (35.1)	100 (+1)
DeepWalk$_{mid}$	24.0 (**27.1**)	31.0 (33.8)	32.8 (35.7)	225 (+1)
DeepWalk$_{high}$	24.9 (26.4)	31.5 (33.7)	33.7 (35.9)	935 (+1)
LINE$_{1st}$	23.1 (24.7)	29.3 (31.6)	31.8 (33.5)	241 (+1)
LINE$_{2nd}$	21.5 (**25.0**)	27.9 (**31.6**)	30.0 (**33.6**)	244 (+1)
node2vec	25.0 (27.0)	31.9 (34.5)	35.1 (37.2)	454 (+1)

类准确率。根据前人工作[31] 的建议，我们在将网络嵌入提供给分类器前，对网络嵌入的每维进行归一化，使得每维的 $L2$ 范数等于 1。我们同样在 NEU 算法更新的前后进行归一化。实验结果见表2.2，2.3和2.4。括号内的数字代表了 NEU 算法更新后的效果。在运行时间列的 “+0.1’’，“+0.3’’，“+1’’ 和 “+8’’ 代表了 NEU 的额外运行时间。以 Cora 数据集为例，当训练比例为 10% 时，NEU 用 0.1 秒将 TADW 的分类准确率从 78.1 提升到 84.4。我们**加粗**了 NEU 取得超过 10% 相对提升的结果。我们进行了显著性检验（0.05 level paired t-test），并将没有通过检验的结果标记为 *。

2.4.4　链接预测

为了进行链接预测的测试，我们需要在给定网络嵌入的情况下对每对节点进行评分。对每对节点表示 r_i 和 r_j，我们尝试了三个评分函数：余弦相似度 $\frac{r_i \cdot r_j}{||r_i||_2||r_j||_2}$，内积 $r_i \cdot r_j$ 和逆 $L2$-距离 $1/||r_i - r_j||_2$。我们使用 AUC 值[32]，即一条未观测到的边比一条不存在的边的评分高的概率，作为我们的评价指标。我们为每个基线方法选取性能最好的评分函数。我们随机去掉 Cora 的 20% 边，BlogCatalog 和 Flickr 的

表 2.4　Flickr 数据集分类结果

	% Macro-F1			运行时间 (s)
% 训练比例	1%	5%	9%	
GF	4.3 (**5.2**)	4.9 (**5.4**)	5.0 (5.4)	241 (+8)
SC	8.6 (**10.9**)	11.6 (**14.3**)	12.3 (**15.0**)	102 (+8)
DeepWalk$_{low}$	7.8 (**8.6**)	10.1 (**11.6**)	10.4 (**12.1**)	1,449 (+8)
DeepWalk$_{mid}$	8.8 (**9.9**)	12.3 (**14.3**)	13.2 (**15.1**)	2,282 (+8)
DeepWalk$_{high}$	10.5 (**11.6**)	17.1 (17.8)	19.1 (19.8)	9,292 (+8)
LINE$_{1st}$	10.3 (10.7)	16.0 (16.6)	17.6 (18.2)	2,664 (+8)
LINE$_{2nd}$	7.8 (8.5)	13.1 (13.5)	14.7 (15.2)	2,740 (+8)
	% Micro-F1			运行时间 (s)
% 训练比例	1%	5%	9%	
GF	21.1 (21.8)	22.0 (23.1)	21.7 (23.4)	241 (+8)
SC	24.1 (**29.2**)	27.5 (**34.1**)	28.3 (**34.7**)	102 (+8)
DeepWalk$_{low}$	28.5 (**31.4**)	30.9 (33.5)	31.3 (33.8)	1,449 (+8)
DeepWalk$_{mid}$	29.5 (31.9)	32.4 (35.1)	33.0 (35.4)	2,282 (+8)
DeepWalk$_{high}$	31.8 (33.1)	36.3 (36.7)	37.3 (37.6)	9,292 (+8)
LINE$_{1st}$	32.0 (32.7)	35.9 (36.4)	36.8 (37.2)	2,664 (+8)
LINE$_{2nd}$	30.0 (31.0)	34.2 (34.4)	35.1 (35.2)*	2,740 (+8)

50% 边作为测试集，并用剩下的边训练节点表示。我们还添加了三个常用的链接预测方法作为参考：Common Neighbors (CN)，Jaccard Index 和 Salton Index[33]。我们只报告 DeepWalk∈{DeepWalk$_{low}$, DeepWalk$_{mid}$, DeepWalk$_{high}$} 和 LINE∈{LINE$_{1st}$, LINE$_{2nd}$} 最好的结果。我们省略了 node2vec 的结果，因为其和最好的 DeepWalk 效果相当或更差。实验结果如图2.1所示。对于每个数据集，最左边三列是传统的三个链接预测基线，之后每对代表一个网络表示学习算法和它被 NEU 增强后的结果。

2.4.5　实验结果分析

我们对两个评估任务的实验结果有四个主要观察结果：

(1) NEU 在两个评估任务上使用几乎可忽略不计的运行时间，一致且显著地提高了各种网络嵌入算法的性能。在 Flickr 数据上的绝对提升没有 Cora 和 BlogCatalog 上面显著，这主要是因为 Flickr 数据集的平均度数有 147，远远大于其他两个数据集。因此高阶邻近度的信息被一阶邻近度信息稀释了。但对于平均度数只有 4 的 Cora 数据集，NEU 具有非常显著的改进，因为高阶邻近度对于稀疏网络起着重要作用。

(2) NEU 可以促使网络表示学习方法更快更稳定的收敛。我们可以看到 Deep-

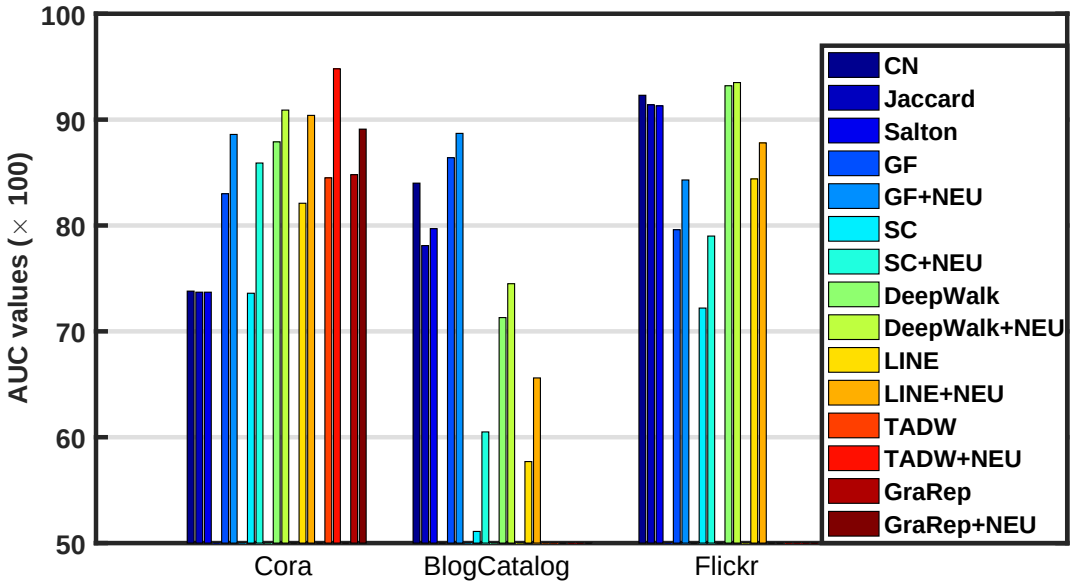


图 2.1　链接预测实验结果

Walk$_{low}$+NEU 和 DeepWalk$_{mid}$+NEU 的效果分别比 DeepWalk$_{mid}$ 和 DeepWalk$_{high}$ 的效果相当甚至更好，而前者却用了更少的总时间。此外，DeepWalk 在 Cora 数据集遇到了过拟合问题：分类准确率随着参数规模增多而减少。但是 DeepWalk+NEU 的效果却十分鲁棒和稳定。

(3) NEU 对于不能归于我们两步框架的 node2vec 算法同样有效。NEU 不会降低 node2vec 的性能，而且略有提升。这个观察证明了 NEU 的有效性和鲁棒性。

(4) NEU 可以作为评估未来网络表示学习方法的预处理步骤，因为 NEU 不会增加时间和空间复杂度。

2.5　本章小结

在本章中，我们提出了一个统一的网络表示学习框架，该框架涵盖了许多现有的网络表示学习方法。我们分析了这个框架的第一步，即邻近度矩阵构建，对比了不同网络表示学习方法使用的邻近度矩阵，并得出结论：如果额外的更高阶邻近度信息被合适地编码进邻近度矩阵，我们可以学习得到更好的网络嵌入表示。然后我们提出网络嵌入更新（NEU）算法，通过隐式逼近高阶邻近度矩阵来改善任何给定网络嵌入的性能。NEU 的运行时间几乎可以忽略不计，相对于基线方法的改进是一致且显著的。

第 3 章　结合富特征信息的网络表示学习

表示学习已经在许多任务中显示出它的有效性，例如图像分类和文本挖掘。网络表示学习旨在学习网络中每个节点的分布式向量表示，已经被认为是网络数据分析的一个重要方面。大多数网络表示学习方法仅使用网络结构信息学习网络表示。实际上，网络中的节点往往包含着丰富的特征信息（例如文本特征），但是这些信息不能很好地应用在当前的表示学习算法框架内。在本章[①]中，我们受第2章工作的启发，在等价的矩阵分解（MF）形式的 DeepWalk（一种最先进的网络表示方法）的基础上，提出了以文本特征为例的结合富特征信息的 DeepWalk（TADW）算法用于网络表示学习。TADW 在矩阵分解的框架下将节点的文本特征结合到网络表示学习中。我们通过将学习得到的节点表示应用于节点多分类任务来评估该方法和各种基线方法。实验结果表明，该方法在所有三个数据集上均优于其他基线，特别是当网络结构噪声较大或者训练数据比例比较小时。

3.1　问题描述

网络数据在人们的日常生活中无处不在，例如 Facebook 用户之间的好友关系或者学术论文之间的引用关系。近年来，研究者们对网络数据中许多重要的机器学习应用进行了广泛的研究，比如节点分类[34]，标签推荐[9]，异常检测[35] 和链接预测[36]。数据稀疏性是这些任务面临的主要问题。为了解决这个问题，网络表示学习（NRL）在统一的低维空间中对每个节点进行编码和表示。网络表示学习有助于我们更好地理解节点之间的结构相关性，进一步减轻数据稀疏性带来的不便[1]。

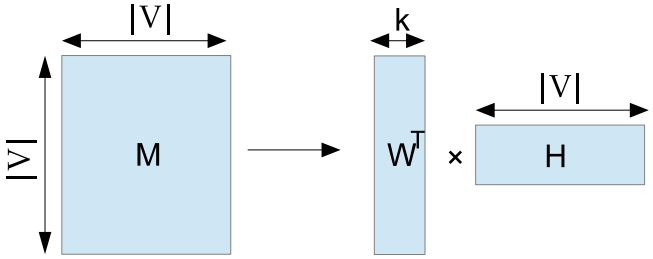
现有的网络表示学习算法主要从网络结构中学习节点表示。举例来说，social dimensions[2,29] 通过计算网络的拉普拉斯矩阵或者模块度矩阵的特征向量来获得节点表示。最近，自然语言处理领域的词表示模型 Skip-Gram 被应用在网络中的随机游走序列上来学习网络中的节点表示，称为 DeepWalk 算法[1]。Social dimensions 和 DeepWalk 都将网络结构作为输入来学习节点表示，而没有考虑任何的外部信息。

在现实世界中，网络中的节点通常具有丰富的信息，例如文本内容和其他元数据。例如，维基百科文章彼此连接并形成网络，并且每篇文章作为节点具有实质文本信息，这对于网络表示学习也是潜在的可利用信息。因此，我们以文本特

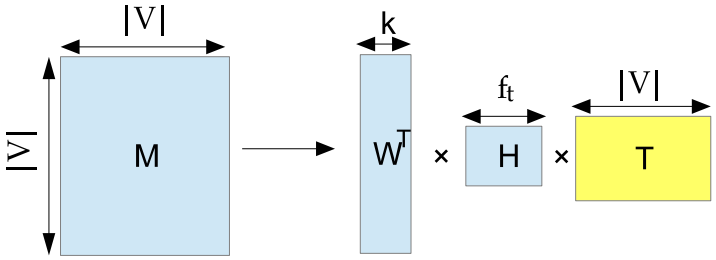
① 本章主要工作以 “Network Representation Learning with Rich Text Information’’ 为题发表在 2015 年的国际学术会议 “The International Joint Conference on Artificial Intelligence (IJCAI’15)’’ 上。

征为例，提出了一个网络结构和特征信息相结合学习网络表示的想法。

一种简单直接的方法是独立地从文本特征和网络特征中学习表示，然后将两种表示拼接起来。然而，该方法没有考虑网络结构和文本信息之间的复杂关系，因此通常效果一般。然而将文本信息纳入现有的网络表示学习框架的方法并不显而易见。例如，DeepWalk 在网络中随机游走时无法简单处理其他来源的信息。



(a) 矩阵分解形式的 DeepWalk



(b) 结合文本信息的矩阵分解 (TADW)

图 3.1　矩阵分解形式的 DeepWalk 和 TADW 示意图。

幸运的是，给定网络 $G = (V, E)$，第2章的工作证明了 DeepWalk 实际上等价于分解矩阵 $M \in \mathbb{R}^{|V|\times|V|}$，其中每个元素 M_{ij} 是节点 v_i 在固定步数内随机游走到 v_j 的平均概率的对数。图3.1(a) 展示了矩阵分解形式的 DeepWalk：将矩阵 M 分解成两个低维矩阵 $W \in \mathbb{R}^{k\times|V|}$ 和 $H \in \mathbb{R}^{k\times|V|}$ 的乘积，其中 $k \ll |V|$。DeepWalk 将矩阵 W 作为 k 维的节点表示。

矩阵分解形式的 DeepWalk 启发我们在矩阵分解的框架下引入文本信息。图 3.1(b) 展示了本章工作的主要思路：把矩阵 M 分解成三个矩阵 $W \in \mathbb{R}^{k\times|V|}$，$H \in \mathbb{R}^{k\times f_t}$ 和文本特征 $T \in \mathbb{R}^{f_t\times|V|}$ 的乘积，然后将 W 和 HT 拼接成 $2k$ 维的节点表示。

本章工作在三个数据集上的对提出的算法和基线方法进行了测试。当训练比例从 10% 到 50% 时，本工作学习得到的表示的分类准确率超过基线方法 2% 到 10%。当训练比例小于 10% 时，我们也使用半监督分类器 Transductive SVM (TSVM)

做了测试。当训练比例为 1% 时，本章工作的方法比其它基线方法有 5% 到 20% 的提升，特别是网络结构信息噪声较大的时候。

总结来说，本章工作有以下两个主要的贡献：

(1) 本章工作首次提出将文本特征引入到网络表示学习中，并提出了基于矩阵分解的 TADW 算法。

(2) 和基线方法相比，本章提出的 TADW 模型取得了 5% 到 20% 的效果提升，特别是训练比例小的情况下。

3.2　相关工作

表示学习技术被广泛应用于计算机视觉[37] 和自然语言处理[3] 等领域。现有的网络表示学习工作[1,2,29,38] 都只考虑了网络的拓扑结构信息，且无法被简单地推广到处理节点的其它外部特征。具体来讲，DGE[38] 和 Spectual Clustering[29] 先设计了基于网络拓扑结构的最优化目标，然后将其转化为特征向量的计算问题。DeepWalk[1] 先从图中生成随机游走序列，然后用词表示学习模型 Skip-Gram 学习得到网络中节点的向量表示。LINE[5] 以网络中的节点向量表示为参数，对节点间的一阶、二阶邻近度进行建模来学习大规模网络的节点表示。GraRep[6] 首先通过矩阵乘法计算了不同阶的邻近度矩阵，然后用 SVD 分解对其进行降维操作，并把每个邻近度矩阵对应的表示拼接起来作为结果。因为 GraRep 需要进行大规模矩阵乘法的计算，所以其计算效率非常低。

据我们所知，在此之前还没有结合富特征信息的网络表示学习算法。一些主题模型（topic model），例如 NetPLSA[39] 在训练时同时考虑了网络结构和文本信息，即在主题模型之外根据网络拓扑结构关系添加额外的正则项。对于 NetPLSA 模型，我们可以将节点的主题分布当作节点表示，并用作本章工作的基线模型。

3.3　模型框架

在这一节中，我们首先形式化定义网络表示学习，然后在矩阵分解框架的基础上提出结合文本特征的网络表示学习算法 TADW。

3.3.1　问题定义

网络表示学习的形式化定义如下：给定网络 $G = (V, E)$，我们想要为网络中每个节点 v 学习一个低维的向量表示 $r_v \in \mathbb{R}^k$，其中维度 k 应远小于节点数 $|V|$。

作为稠密的实数值表示，r_v 可以减轻网络表示的稀疏性问题（例如邻接矩阵）。我们可以将 r_v 视为节点 v 的特征。这些特征可以方便地作为例如逻辑回归和支持

向量机（SVM）等分类器的输入，并应用于节点分类等机器学习任务。需要注意的是，我们学习得到的网络表示不是针对特定任务的，而是可以在不同的任务之间通用。

根据第2章中 DeepWalk 等效于矩阵分解的证明，我们提出基于 DeepWalk 派生的矩阵分解框架来向网络表示学习引入文本信息。下面，我们先对低秩矩阵分解进行简单的介绍，然后正式介绍我们结合了网络和文本信息的表示学习算法。

3.3.2　低秩矩阵分解

矩阵是表示关系数据的常用方法。矩阵分析的一个问题是通过其一小部分元素来推理矩阵的内在结构。一个通常的假设是矩阵 $M \in \mathbb{R}^{b\times d}$ 近似地拥有较低的秩 k，其中 $k \ll \{b,d\}$。基于此假设，我们可以用低秩近似来补全矩阵 M 中的缺失元素。然而，解决一个秩约束的最优化问题一般是 NP 难的。因此，研究者们转而寻找矩阵 $W \in \mathbb{R}^{k\times b}$ 和 $H \in \mathbb{R}^{k\times d}$ 来最小化损失函数 $L(M, W^T H)$ 和一个迹范数约束。，这个约束被进一步替换成了一个更简单的惩罚项[40]。在本章的工作中，我们使用平方损失函数。

正式地，我们让矩阵 M 中被观测到的集合为 Ω。我们想要找到矩阵 $W \in \mathbb{R}^{k\times b}$ 和 $H \in \mathbb{R}^{k\times d}$ 来最小化

$$\min_{W,H} \sum_{(i,j)\in\Omega} \left(M_{ij} - (W^T H)_{ij}\right)^2 + \frac{\lambda}{2}\left(||W||_F^2 + ||H||_F^2\right), \tag{3-1}$$

其中 $||\cdot||_F$ 表示矩阵的 Frobenius 范数，λ 是平衡两部分的调节参数。

低秩矩阵分解只基于 M 的低秩假设来补全矩阵 M。如果矩阵 M 中的对象拥有额外的特征，我们可以使用归纳矩阵补全[41] 来利用这些特征。通过将两个特征矩阵合并到目标函数中，归纳矩阵补全可以利用行和列单元的更多信息。假设我们有特征矩阵 $X \in \mathbb{R}^{f_x\times b}$ 和 $Y \in \mathbb{R}^{f_y\times d}$，其中 X 和 Y 的第 i 列分别是单元 i 的 f_x 和 f_y 维的向量特征。我们的目标是计算矩阵 $W \in \mathbb{R}^{k\times f_x}$ 和 $H \in \mathbb{R}^{k\times f_y}$ 来最小化

$$\min_{W,H} \sum_{(i,j)\in\Omega} \left(M_{ij} - (X^T W^T H Y)_{ij}\right)^2 + \frac{\lambda}{2}\left(||W||_F^2 + ||H||_F^2\right). \tag{3-2}$$

需要注意的是，归纳矩阵补全是为了利用特征来完成矩阵补全[41]，目标和我们的工作完全不同。受归纳矩阵补全工作的启发，我们将文本信息引入网络表示学习。

3.3.3　文本辅助的 DeepWalk (TADW)

给定网络 $G = (V, E)$ 和对应的文本特征矩阵 $T \in \mathbb{R}^{f_t \times |V|}$，我们提出 Text-Associated DeepWalk (TADW) 来从网络结构 G 和文本特征 T 中学习每个节点 $v \in V$ 的表示。

我们在第2章中证明了 DeepWalk 等价于分解了矩阵 $M = \log \frac{A+A^2+\cdots+A^t}{t}$，其中 t 是 Skip-Gram 的窗口大小，A 是行归一化的邻接矩阵，其每行的和都等于 1。当 t 变大时，计算精确的 M 具有 $O(|V|^3)$ 的复杂度。实际上，DeepWalk 使用基于随机游走的采样方法来避免显式计算精确的矩阵 M。当 DeepWalk 对更多随机游走进行采样时，性能会更好，而算法效率会降低。

在 TADW 中，我们找到了一个速度和效果的平衡：分解矩阵 $M = (A + A^2)/2$。这里我们为了计算效率分解 M 而不是 $\log M$：因为 $\log M$ 比 M 有远远更多的非零元素，而平方损失函数下的矩阵分解的计算复杂度正比于矩阵 M 中的非零元素数量[40]。因为现实中大多数网络是稀疏的，也就是说 $O(E) = O(V)$，计算矩阵 M 需要 $O(|V|^2)$ 时间。如果网络很稠密，我们甚至可以直接分解矩阵 A。我们的目标是求解矩阵 $W \in \mathbb{R}^{k \times |V|}$ 和 $H \in \mathbb{R}^{k \times f_t}$ 来最小化

$$\min_{W,H} ||M - W^T HT||_F^2 + \frac{\lambda}{2}(||W||_F^2 + ||H||_F^2). \tag{3-3}$$

为了优化求解 W 和 H，我们迭代地最小化 W 和 H，因为对单一矩阵 W 或 H 的优化是凸优化。虽然 TADW 可能会收敛到局部最小值而不是全局最小值，但如我们的实验所示，我们的方法在实践中运行良好。

和专注于补全矩阵 M 的低秩矩阵分解和归约矩阵补全不同，TADW 的目标是引入文本特征学习更好的网络表示。另外，归约矩阵补全直接从原始数据获取矩阵 M，而我们从矩阵分解形式的 DeepWalk 算法推导中构建矩阵 M。因为 TADW 得到的 W 和 HT 都可以被看作节点的 k 维表示，我们将他们拼接起来作为统一的 $2k$ 维网络表示。在实验中，我们将证明该统一表示明显优于网络表示和文本特征（矩阵 T）的简单拼接。

3.3.4　复杂度分析

在 TADW 中，计算矩阵 M 需要 $O(|V|^2)$ 时间。我们使用先前工作[40] 中的快速优化算法来求解公式 (3-3) 中的最优化问题。每轮迭代中，最小化 W 和 H 的时间复杂度是 $O(\text{nnz}(M)k + |V|f_t k + |V|k^2)$，其中 $\text{nnz}(\cdot)$ 表示非零元素数量。作为比较，传统矩阵分解的复杂度，即公式 (3-1) 中的最优化问题，是 $O(\text{nnz}(M)k + |V|k^2)$。在

我们的实验中，算法可以在 10 轮迭代内收敛。

3.4　实验结果

我们使用多类节点分类任务来评价网络表示的质量。正式地，我们将学习得到的网络表示 $\mathcal{R} = \{r_1, r_2, \ldots, r_{|V|}\}$ 看作节点的特征。我们的任务是根据节点特征 $\mathcal{R}$ 和标注集 L 预测未标注集 U 的标签。

机器学习中的许多分类器可以处理这个任务。我们分别选择 SVM 和 transductive SVM 进行监督和半监督训练和测试。注意，由于网络表示学习过程没有使用训练集中的节点标签，因此是完全无监督的。

我们在三个公开可用的数据集上，使用五种基线表示学习方法来评估 TADW 的效果。我们从文档之间的链接或引用以及这些文档的词频-逆文本频率（TFIDF）矩阵中学习表示。

3.4.1　数据集

Cora 包含来自 7 个分类的 2,708 篇机器学习论文和 5,429 条它们之间的链接。这些链接是文档之间的引用关系。每篇文章拥有 1,433 维的二进制文本特征向量，表示对应单词是否出现在文章之中。

Citeseer 包含来自 6 个分类的 3,312 篇论文和 4,732 条链接。和 Cora 类似，这些链接也对应着文章间的引用关系。每篇文章有 3,703 维的二进制文本特征向量。

Wiki 包含来自 19 个分类的 2,405 篇文档和 17,981 条文档之间的超链接。这个数据集的 TFIDF 矩阵有 4,973 列。

Cora 和 Citeseer 中的文档是由标题和摘要生成的短文本。我们剔除了停止词和文档频率低于 10 次的单词。处理后的 Cora 和 Citeseer 平均一个文档分别含有 18 和 32 个单词。Wiki 中的文档是长文本，每篇文档平均含有 640 个单词。我们去掉了和其它文档没有链接的文档，并将所有网络视为无向图。

3.4.2　TADW 设置

对于全部三个数据集，我们通过对 TFIDF 矩阵的 SVD 分解把文本特征的维度减到 200 来获取文本特征矩阵 $T \in \mathbb{R}^{200\times|V|}$。这个预处理可以减少参数矩阵 H 的规模。我们也将文本特征 T 作为一个只考虑文本信息的基线方法。我们对 Cora 和 Citeseer 数据集选取 $k = 80$，$\lambda = 0.2$; 对 Wiki 数据集 $k = 100,200$，$\lambda = 0.2$。注意 TADW 的向量表示维度是 $2k$。

3.4.3　基线方法

DeepWalk. DeepWalk[1] 是只考虑网络结构的网络表示学习算法。参数设置如下：每个节点随机游走数 $\gamma = 80$，窗口大小 $t = 10$，和原始论文保持一致。表示维度选取了从 50 到 200 间的最佳值：对于 Cora 和 Citeseer 数据集 $k = 100$，对于 Wiki 数据集 $k = 200$。通过求解公式 (3-1) 并拼接 W 和 H 作为节点表示，我们也测试了矩阵分解形式的 DeepWalk。其表现和原始 DeepWalk 相当，所以我们只展示了原始 DeepWalk 的结果。

PLSA. 我们使用 PLSA[42] 从 TFIDF 矩阵训练主题模型。PLSA 是一个只考虑文本的基线方法。PLSA 利用 EM 算法估计文档和单词的主题分布。我们使用文档的主题分布作为节点表示。

Text Features. 我们使用文本特征矩阵 $T \in \mathbb{R}^{200\times|V|}$ 作为 200 维的表示。这个方法同样是只考虑文本的基线。

Naive Combination. 我们直接将 Text Features 和 DeepWalk 的向量前后拼接起来。该基线方法对于 Cora 和 Citeseer，维度是 300；对于 Wiki，维度是 400。

NetPLSA.[39] 提出将文档间的链接作为网络正则化项来学习文档的主题模型。其基本假设是相连的文档应具有类似的主题分布。我们将结合了网络结构的文档主题分布作为节点表示。主题模型 NetPLSA 可以被看作同时考虑了网络和文本的网络表示学习算法。对于 Cora 和 Citeseer，我们设置主题数为 160，对于 Wiki 为 200。

3.4.4　分类器和实验设置

对于监督分类器，我们使用 Liblinear[30] 实现的线性 SVM。对于半监督分类器，我们使用 SVM-Light[43] 实现的 transductive SVM （TSVM）。对于 TSVM 我们使用线性核。我们对每个分类训练了一对多的分类器并选取得分最高的分类作为预测结果。

我们将节点表示作为特征来训练分类器，并使用不同的训练比率来评估分类准确性。对于监督训练的线性 SVM，训练比率从 10% 变化到 50%；对于半监督训练的 TSVM，训练比率从 1% 变化到 10%。对于每个训练比率，我们随机选择文档作为训练集，其余文档作为测试集。我们重复 10 次并报告平均准确率。

3.4.5　实验结果分析

表 3.1，表 3.2和表 3.3展示了 Cora，Citeseer 和 Wiki 数据集上的分类准确率。这里 “-” 表示 TSVM 因为特征质量低，无法在 12 小时内收敛 (TSVM 对 TADW 可

表 3.1　Cora 数据集实验结果.

分类器	Transductive SVM				SVM				
% 训练比例	1%	3%	7%	10%	10%	20%	30%	40%	50%
DeepWalk	62.9	68.3	72.2	72.8	76.4	78.0	79.5	80.5	81.0
PLSA	47.7	51.9	55.2	60.7	57.0	63.1	65.1	66.6	67.6
Text Features	33.0	43.0	57.1	62.8	58.3	67.4	71.1	73.3	74.0
Naive Combination	67.4	70.6	75.1	77.4	76.5	80.4	82.3	83.3	84.1
NetPLSA	65.7	67.9	74.5	77.3	80.2	83.0	84.0	84.9	85.4
TADW	**72.1**	**77.0**	**79.1**	**81.3**	**82.4**	**85.0**	**85.6**	**86.0**	**86.7**

表 3.2　Citeseer 数据集实验结果.

分类器	Transductive SVM				SVM				
% 训练比例	1%	3%	7%	10%	10%	20%	30%	40%	50%
DeepWalk	-	-	49.0	52.1	52.4	54.7	56.0	56.5	57.3
PLSA	45.2	49.2	53.1	54.6	54.1	58.3	60.9	62.1	62.6
Text Features	36.1	49.8	57.7	62.1	58.3	66.4	69.2	71.2	72.2
Naive Combination	39.0	45.7	58.9	61.0	61.0	66.7	69.1	70.8	72.0
NetPLSA	45.4	49.8	52.9	54.9	58.7	61.6	63.3	64.0	64.7
TADW	**63.6**	**68.4**	**69.1**	**71.1**	**70.6**	**71.9**	**73.3**	**73.7**	**74.2**

表 3.3　Wiki 数据集实验结果.

分类器	SVM						
% 训练比例	3%	7%	10%	20%	30%	40%	50%
DeepWalk	48.4	56.6	59.3	64.3	66.2	68.1	68.8
PLSA	58.3	66.5	69.0	72.5	74.7	75.5	76.0
Text Features	46.7	60.8	65.1	72.9	75.6	77.1	77.4
Naive Combination	48.7	62.6	66.3	73.0	75.2	77.1	78.6
NetPLSA	56.3	64.6	67.2	70.6	71.7	71.9	72.3
TADW (k=100)	**59.8**	**68.2**	**71.6**	**75.4**	**77.3**	**77.7**	**79.2**
TADW (k=200)	**60.4**	**69.9**	**72.6**	**77.3**	**79.2**	**79.9**	**80.3**

以在五分钟内收敛)。我们没有展示 Wiki 数据集上的半监督实验结果，因为监督训练的 SVM 已经在该数据集上以较小的训练比率获得了相近甚至更好的性能。所以对于 Wiki 数据集，我们只展示监督训练的结果。Wiki 相较于其他两个数据有更多的类别，也就需要更多数据来做充分的训练，所以我们把最小训练比例定为 3%。从这些实验结果表格中，我们有以下观察：

(1) TADW 在全部三个数据集上一致超过其他基线方法。更进一步，TADW 在 Cora 和 Citeseer 数据集上可以在少用 50% 训练数据的情况下打败其他方法。这些

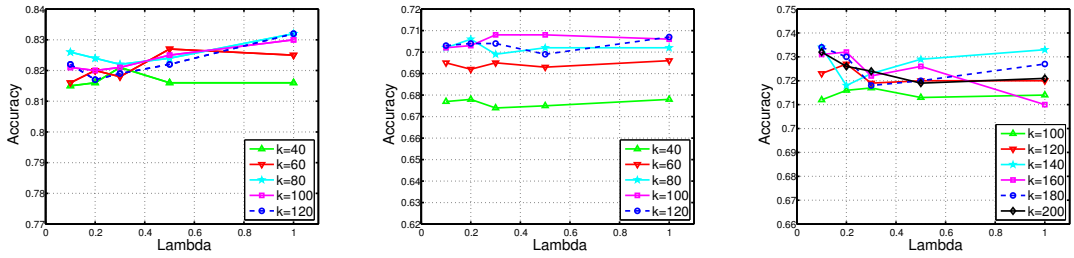
实验证明了 TADW 算法的有效性和鲁棒性。

(2) TADW 在半监督学习中有更加显著的提升。TADW 在 Cora 数据集上超过最好的基线方法 4%，Citeseer 数据集上 10% 到 20%。这主要是因为 Citeseer 数据中单纯利用网络结构信息的表示质量较差。相比于 DeepWalk 和文本特征的简单组合，TADW 在从含噪音的数据中学习的效果更加鲁棒。

(3) TADW 在训练比率低的时候提升显著。大多数基线方法的准确率随着训练比率降低迅速下降，因为它们的节点表示噪音更大。相反，因为 TADW 统一地从网络和文本信息中学习表示，其表示中的噪音更小且更一致。

这些观察证明了 TADW 算法产生的表示的质量很高。此外，TADW 是任务无关的算法，学习得到的表示可以方便的用于各种不同的任务中，如链接预测、相似度计算和节点分类。TADW 的分类准确率和一些集体分类算法[44–46] 的效果相当，即使我们在训练中没有针对分类问题做出针对性的优化。

3.4.6　参数敏感性



(a) Cora　　(b) Citeseer　　(c) Wiki

图 3.2　k 和 λ 的参数敏感性实验

TADW 有两个超参数：维度 k 和正则项权重 λ。我们将训练比率固定在 10%，并测试了不同 k 和 λ 设置下的分类准确率。

对于 Cora 和 Citeseer 数据集，我们让 k 从 40 变化到 120，λ 从 0.1 变化到 1；对于 Wiki 数据集，k 从 100 变化到 200，λ 从 0.1 到 1。图 3.2 展示了不同 k 和 λ 下准确率的变化情况。在 Cora，Citeseer 和 Wiki 数据集上，对于固定的 k，准确率的变化分别在 1.5%，1% 和 2% 以内。对于 Cora 和 Citeseer 数据集，$k \geq 80$；对于 Wiki，$k \geq 140$ 时准确率基本相当。因此，在 k 和 λ 在合理区间内变化时，TADW 效果稳定。

表 3.4　DeepWalk 和 TADW 对应的五篇最相似文档

DeepWalk 的五篇最相似文档	
标题	类别标签
Feature selection methods for classifications	Neural Network
Automated model selection	Rule Learning
Compression-Based Feature Subset Selection	Theory
Induction of Condensed Determinations	Case Based
MLC Tutorial A Machine Learning library of C classes	Theory
TADW 的五篇最相似文档	
标题	类别标签
Feature subset selection as search with probabilistic estimates	Theory
Compression-Based Feature Subset Selection	Theory
Selection of Relevant Features in Machine Learning	Theory
NP-Completeness of Searches for Smallest Possible Feature Sets	Theory
Feature subset selection using a genetic algorithm	Genetic Algorithms

3.4.7　案例分析

为了更好地理解文本信息对于网络表示学习的有效性，我们展示了 Cora 数据集上的一个例子。对应论文标题为 “Irrelevant Features and the Subset Selection Problem’’，简称为 IFSSP。这篇文章的标签是 “Theory’’。如表 3.4所示，利用 DeepWalk 和 TADW 产生的表示，我们根据余弦相似度排序分别找了和 IFSSP 最相似的 5 篇文档。

我们发现找到的所有论文都被 IFSSP 引用。然而，DeepWalk 找到的 5 篇文档中，有 3 篇有不同的类别标签，而 TADW 找到的前 4 篇文档都有同样的 “Theory’’ 标签。这表明相比于纯粹基于网络结构的 DeepWalk，TADW 在文本信息的帮助下可以学习到更好的网络表示。

DeepWalk 找到的第 5 篇文档同时展示了只考虑网络结构的另一个局限性。“MLC Tutorial A Machine Learning library of C classes” (简称为 MLC) 是描述一个广泛使用的工具包的文档，可能会被许多不同领域的工作引用。一旦这些工作中的

一部分也引用了 IFSSP，DeepWalk 会倾向于让 IFSSP 和 MLC 有类似的向量表示，即使它们完全属于不同的领域。

3.5　本章小结

在本章中，我们以文本特征为例，从矩阵分解的角度出发，提出了网络表示学习算法 TADW 来引入富特征信息。在三个数据集上，不同训练比率下的实验结果验证了 TADW 算法的有效性和鲁棒性。TADW 也可以被看作是融合两种来源信息的框架：相比于简单的将两种特征拼接起来，TADW 提供了一种通过矩阵分解联合建模的特征组合的全新角度。

第 4 章　富信息网络典型应用问题——基于位置的社交网络的推荐系统

在前面的章节中，本文主要对缺乏对于已有网络表示学习算法的理论分析和现有网络表示学习方法忽略了网络拓扑结构以外的丰富信息的问题做出了改进。在本章及后续章节中，本文将以网络表示学习技术作为模型的底层，并根据特定的富信息网络场景利用包括循环神经网络、卷积神经网络在内的深度学习模型进行建模，在推荐系统和传播预测等重要应用任务中解决现有工作难以应用于相对复杂的典型应用问题的缺点。

移动轨迹数据在基于位置的服务中的加速增长为理解用户的移动行为带来了宝贵的数据资源。除了记录轨迹数据之外，这些基于位置的服务的另一个主要特征是它们还允许用户添加他们喜欢或感兴趣的任何人。社交网络和基于位置的服务的组合被称为基于位置的社交网络（LBSN）。根据前人工作[47] 的结论，社交相关的用户经常访问的位置往往也是相关的，这表明社交关系和 LBSN 中用户的轨迹行为之间存在着密切关联。为了更好地分析和挖掘 LBSN 数据，我们需要从一个全面的视角来同时分析和挖掘来自社交网络和移动轨迹数据这两个方面的信息。

在本章[①]中，我们提出了一种可以同时建模社交网络和移动轨迹的神经网络模型。模型由两部分组成：社交网络的构建和移动轨迹的生成。首先，模型为每个用户学习网络表示，并利用网络表示来构建社交网络。模型的关键在于产生移动轨迹的部分。其次，模型考虑了影响移动轨迹生成过程的四个因素，即用户访问偏好，好友关系影响，短期序列上下文和长期序列上下文。为了刻画两个上下文，模型分别使用循环神经网络和带门循环单元来捕获不同级别（即短期或长期）的移动轨迹中的序列相关性。最后，整体模型通过共享的用户网络表示参数来联合这两个部分。两个重要应用的实验结果证明了该模型的有效性。特别是在网络结构或轨迹数据稀疏时，该算法对基线的改进更为显著。

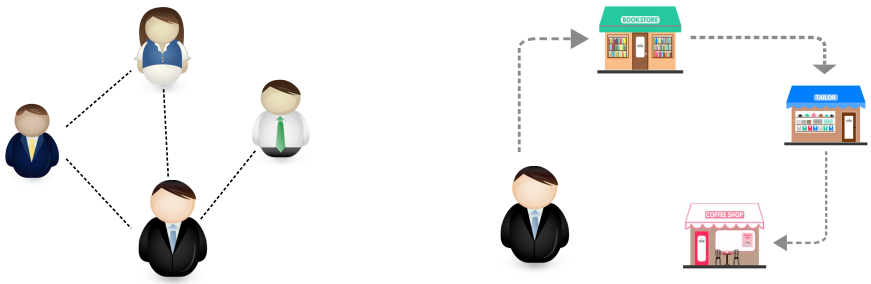
4.1　问题描述

近年来，移动设备（例如智能手机和平板电脑）的使用几乎无处不在。随着互联网技术的创新和发展，移动设备已成为用户连接广阔世界中在线信息的重要渠道。在日常生活中，用户可以使用智能手机开展许多活动，包括研究旅行计划，访

① 本章主要工作以 “A Neural Network Approach to Jointly Modeling Social Networks and Mobile Trajectories’’ 为题发表在 2017 年的国际学术期刊 “ACM Transactions on Information Systems (ACM TOIS)’’ 上。

问在线教育和寻找工作。移动端使用的加速增长为数据挖掘研究带来了海量的数据。在这些丰富的移动数据中，一种重要的数据资源是从移动设备上的 GPS 传感器获得的大量移动轨迹数据。这些传感器足迹为用于发现用户的轨迹模式并了解其移动行为的研究提供了宝贵的信息资源。一些基于位置的共享服务已经出现并受到很多关注,，例如 Gowalla①和 Brightkite②。

除了记录用户轨迹数据之外，这些基于位置的服务的另一个主要特征是它们还允许用户添加他们喜欢或感兴趣的任何人。以 Brightkite 为例，你可以使用手机的内置 GPS 跟踪你的朋友或附近的任何其他 Brightkite 用户。社交网络和基于位置的服务的组合产生了一种特定的社交网络风格，称为基于位置的社交网络(LBSN)[48–50]。图4.1展示了 LBSN 的示意图。可以看到，LBSN 通常包括社交网络和移动轨迹数据。最近的文献表明，社交链接信息对于改进现有的推荐任务非常有用[51–53]。直觉上看，经常访问相同或相似位置的用户很可能是社交好友且社交好友更可能会访问相同或类似的位置。具体地，一些研究发现 LBSN 中社交关系与用户的轨迹行为之间存在密切联系。一方面，Cho 等人[47] 发现社交相关的用户经常访问的位置往往也是相关的。另一方面，轨迹相似性可用于推断用户之间的社交关系强度[54–56]。因此，我们需要一个从两个方面同时分析和挖掘信息的全新视角。本章工作的目标是通过刻画社交网络和移动轨迹数据来开发一种建模 LBSN 数据的联合模型。



(a) 社交网络　　(b) 用户轨迹

图 4.1　LBSN 数据的示意图：(a) 用户间的连接表示好友关系；(b) 用户生成的轨迹是按时间顺序排列的位置记录序列。

在第一个方面，社交网络分析在过去十年中引起了越来越多的关注。它根据节点（网络中的个体参与者，人或事物）以及连接它们的关系或边（关系或交互）

① https://en.wikipedia.org/wiki/Gowalla
② https://en.wikipedia.org/wiki/Brightkite

来刻画网络结构。人们已经在社交网络上开发了各种应用，包括网络分类[34]、链接预测[11]、异常检测[57] 和社区发现[58]。其中一个基本问题是如何表示网络节点。最近，研究者们提出了网络嵌入模型[1] 来解决网络中的数据稀疏性。在第二个方面，基于位置的服务为用户提供了一种记录其轨迹信息的便捷方式。不同于一般的社交网络分析，学者们已经构建了许多针对性的研究来改进基于位置的服务。典型的应用任务是位置推荐，其目的在于推断用户的访问偏好并为用户提供有意义的推荐。它可以分为三种不同的设置：一般位置推荐[59–61]，时间已知的位置推荐[62–64] 和下一个位置推荐[65–67]。一般位置推荐将生成用户访问位置的总体推荐列表; 而时间已知或下一个位置推荐通过指定时间段或序列性预测进一步对推荐任务施加时间约束。

这两个方面刻画了 LBSN 上的不同数据特征并且彼此相关[60,68]。为了进行更好和更有效的数据分析和挖掘研究，我们需要同时刻画 LBSN 上的网络结构和轨迹行为来开发联合模型。但是，这样的任务是具有挑战性的。社交网络和移动轨迹是异构数据类型。社交网络通常以一张图来描述，而轨迹通常被建模为记录序列。将社交关系结合到应用系统（例如推荐系统）中的常用方式是通过假设链接表现了用户相似性而采用的正则化技术。通过这种方式，社交关系被用作了辅助信息，但却没有被联合模型建模，并且模型性能高度依赖于“同质性原则”。本章工作首次使用神经网络方法联合建模社交网络和移动轨迹。我们的方法受到最近深度学习进展的启发。与其他方法相比，神经网络模型可以作为一种有效的通用函数逼近机制，能够捕捉复杂的数据特征[69]。具体而言，最近的研究表明神经网络模型处理网络和序列数据的优越性。一方面，一些研究试图将网络的节点嵌入到低维向量空间中[1,2,5]，称为网络嵌入。利用这种低维稠密向量，可以减轻稀疏网络表示所遇到的数据稀疏性问题。另一方面，神经网络模型是功能强大的数据计算模型，能够捕捉和表示复杂的输入/输出关系。特别地，研究者们已经提出了几种用于处理序列数据的神经网络模型，例如循环神经网络（RNN）[70]。RNN 和其变体 LSTM 与 GRU 已经在许多应用中表现出了优异的效果。

通过结合网络嵌入和深度学习的序列建模的优点，本章工作提出了一种可以共同建模社交网络和移动轨迹的神经网络模型。具体而言，模型由两部分组成：社交网络的构建和移动轨迹的生成。模型首先采用网络嵌入方法来构建社交网络：为每个用户学习一个节点表示。模型的关键在于产生移动轨迹的部分。模型考虑了影响移动轨迹生成过程的四个因素，即用户访问偏好，好友影响，短期序列上下文和长期序列上下文。前两个因素主要与用户自身有关，而后两个因素主要反映历史轨迹的序列特征。模型设置了两个不同的用户表示来模拟前两个因素：访问兴

趣表示和网络表示。为了刻画最后两个上下文表示，模型使用 RNN 和 GRU 模型来捕捉不同级别（即短期或长期）移动轨迹的序列相关性。最后，我们通过共享用户网络表示参数来联合这两个部分：来自网络结构的信息被编码在用户网络表示中，并随后用于移动轨迹的生成过程。

为了证明所提出模型的有效性，本章工作使用真实数据集，在下一个位置推荐和好友推荐这两个重要的 LBSN 应用上评估模型。对于第一项任务，轨迹数据是主要信息，而网络结构用作辅助数据。本章工作提出的方法一致地优于多个基线方法。实验发现对于轨迹数据很少的用户来说，辅助数据（即网络结构）变得更加重要。对于第二个任务，网络数据是主要信息，而轨迹数据用作辅助数据。与第一项任务类似：本章工作提出的方法依然表现最佳，特别是对于那些好友数量很少的用户。两个重要应用的实验结果证明了该模型的有效性。在本章工作提出的方法中，网络结构和轨迹信息相互补充。因此，当网络结构或轨迹数据稀疏时，整体模型对基线的改进更为显著。

总结来说，本章工作有以下三个主要贡献：

(1) 本章工作提出了一种共同刻画社交网络结构和用户的轨迹行为的神经网络模型。在我们的方法中，网络结构和轨迹信息相互补充，提供了一种建模 LBSN 的异构数据类型的有效方法。

(2) 本章工作提出的模型考虑了移动轨迹生成中的四个因素，包括用户访问偏好，好友影响，短期序列上下文和长期序列上下文。前两个因素由用户的两个不同嵌入表示建模，并进一步采用 RNN 和 GRU 模型来捕捉短期和长期序列上下文。

(3) 两个重要应用的实验结果证明了该模型的有效性。当网络结构或轨迹信息稀疏时，整体模型对基线的改进更为显著。

4.2　相关工作

本章工作主要涉及分布式表示学习，社交链接预测和位置推荐。

4.2.1　分布式表示学习和神经网络模型

基于数据表示学习的机器学习算法在过去几年中取得了巨大成功。数据的表示学习可以为学习分类器和其他预测器提取有用的信息。分布式表示学习已广泛用于许多机器学习任务中[12]，如计算机视觉[37] 和自然语言处理[3]。

在过去十年中，学者们还提出了许多关于网络嵌入学习的工作[1,2,29,38]。传统的网络嵌入学习算法通过计算邻近度矩阵的特征向量来学习节点表示[2,71,72]。例如 DGE[38] 求解拉普拉斯矩阵的广义特征向量；Social dimensions[2] 计算归一化图拉

普拉斯矩阵的前 k 小的特征向量作为 k 维的节点表示。

DeepWalk[1] 在随机游走序列上应用自然语言处理领域一种常用的语言模型 Skip-Gram[3] 学习网络表示。利用深度学习技术进行网络分析的 DeepWalk 比传统的网络表示学习算法更有效，并且可以实现大规模的表示学习。沿着这个方向，LINE[5] 刻画了节点间的一阶和二阶邻近度，GraRep[6] 通过 k 阶邻近度矩阵的 SVD 分解建模局部和全局结构信息。MMDW[73] 将标签信息考虑在内并学习半监督网络嵌入。

TADW 和 PTE[20] 分别扩展了 DeepWalk 和 LINE 来向网络表示学习中引入文本信息。TADW 通过矩阵分解框架将文本信息嵌入到节点表示中，PTE 从异构文本网络中学习半监督嵌入。然而，TADW 和 PTE 都在文档网络上进行实验，并且未考虑单词之间的序列信息。

神经网络模型在过去十年中取得了巨大成功。两个众所周知的神经网络架构是卷积神经网络（CNN）和循环神经网络（RNN）。CNN 用于从各种大小的数据中提取固定长度表示[37]. 针对序列建模的 RNN 及其变体 GRU 已成功应用于句子建模[70]，语音信号建模[74] 和序列点击预测[75]。

4.2.2　社交链接预测

社交链接预测已经通过挖掘图结构模式在各种社交网络中得到广泛研究，例如三角闭合过程[76]，用户统计研究[77]。在本章中，我们主要关注轨迹数据的应用。

研究人员过去经常通过评估序列模式来衡量用户相似度，例如，他们使用一系列停留点来表示用户轨迹并通过序列匹配算法评估用户相似度[78]。为了改进这些方法，人们还考虑了预定义的标签和权重，以更好地刻画停留点[79]。随着 LBSN 越来越受欢迎，轨迹相似性挖掘引起了越来越多的关注。研究者们考虑了许多因素来更好地表征相似性。因此，物理距离[80]，位置类别[81]，空间或时间协同定位率[82] 和时空间约束下的共现[83] 都被用于社交链接预测。位置的流行度和共现关系[54] 被证明在所有因素中是相对重要的。如果使用社交关系来聚类位置，社交关系强度可以反过来由用户共有的聚类来推断[47,56]。

4.2.3　位置推荐

轨迹建模最重要的任务之一是位置推荐。对于一般位置推荐，人们考虑了几种辅助信息，如地理[60,61]，时间[84] 和社交网络信息[68]。为了解决数据稀疏性问题，研究者们还考虑了包括位置类别标签的内容信息[85,86]。位置标签也可被用于概率模型，例如聚合 LDA[87]。文本信息包括文本描述[87–89] 也被用于位置推荐。W^4 Who

(用户)，What (位置类别)，When(时间) 和 Where (位置)[90,91] 对多维协同推荐采用了张量分解。然而，这些工作主要基于协同过滤、矩阵分解或 LDA 的方法，没有对轨迹中的序列信息进行建模。

对于在特定时间推荐位置的时间已知位置推荐任务需要对时间因素进行建模。基于协同过滤的方法[62] 通过线性组合统一了时间和地理信息。地理时间图通过在图上进行偏好传播，用于时间已知位置推荐[63]。此外，时间因素也通过非负矩阵分解[92] 和 RNN[64] 建模。

与一般位置推荐不同，下一个位置推荐还需要考虑当前状态。因此，在下一个位置推荐中考虑序列信息更为重要。大多数之前的工作基于马尔可夫链假设建模序列行为，即假设下一个位置仅由当前位置确定并且独立于先前的位置[65–67,93]。例如 Factorized Personalized Markov Chain (FPMC) 算法[93] 对包含所有用户的转移概率矩阵的三维张量进行分解。Personalized Ranking Metric Embedding (PRME)[94] 通过在两个不同的向量空间中建模用户-位置距离和位置-位置距离来进一步扩展 FPMC。最初用于用户购买行为建模的 Hierarchical Representation Model (HRM)[95] 也可以适用于建模用户轨迹。HRM 构建了一个包含上次交易中的用户和商品特征的双层结构，用于预测下一次交易中的商品。这些方法应用于下一个位置推荐，其旨在给定位置历史记录和用户的当前位置，预测用户将访问的下一个位置。注意马尔可夫链是一个非常强的假设，即假设下一个位置仅由当前位置确定。在实际中，下一个位置也可能受到整个位置历史的影响。

4.3　模型框架

在这一节中，我们首先形式化定义问题，然后提出我们对社交网络和移动轨迹联合建模的模型。

4.3.1　问题定义

令 L 表示位置集合，当用户 v 在时间 s 访问位置 l 时，我们用三元组 $\langle v, l, s\rangle$ 表示这个信息。给定用户 v，其移动轨迹 T_v 是和 v 相关的一系列三元组：$\langle v, l_1, s_1\rangle, ..., \langle v, l_i, s_i\rangle, ..., \langle v, l_N, s_N\rangle$，其中 N 是序列长度且三元组按时间顺序排列。为简洁起见，我们重写了 T_v 的表示为按时间排序的位置序列 $T_v = \{l_1^{(v)}, l_2^{(v)}, \ldots, l_N^{(v)}\}$。此外，我们可以将轨迹分成多个连续的子轨迹：轨迹 T_v 被分成 m_v 个子轨迹 $T_v^1, \ldots, T_v^{m_v}$。每个子轨迹本质上是原始轨迹序列的子序列。为了切分轨迹，我们计算原始轨迹序列中两个访问位置之间的时间间隔，我们和前人工作[65] 一样，当时间间隔大于六小时时进行切分。每个用户对应于包含了多个连续子轨迹 $T_v^1, \ldots, T_v^{m_v}$

的轨迹序列 T_v。T 表示所有用户的轨迹的集合。

除了轨迹数据之外，基于位置的服务也在用户之间提供社交联系。正式地，我们将社交网络建模为图 $G = (V, E)$，其中节点 $v \in V$ 代表一个用户，每条边 $e \in E$ 表示两个用户间的好友关系。在实际应用中，边可以是无向的或有向的。我们的模型可以灵活地处理这两种类型的社交网络。注意这些链接主要反映在线好友关系，并不一定表明两个用户在现实生活中是朋友。

给定社交网络信息 $G = (V, E)$ 和移动轨迹信息 T，我们的目标是开发一个可以建模和利用这两种数据资源的联合模型。这种联合模型应该比仅使用单一数据资源构建的模型更有效。为了测试模型性能，我们在 LBSN 中评估了两个应用任务。

任务 I. 对于下一个位置推荐的任务，我们的目标是为用户 v 推荐其下次可能访问的位置的排序列表。

任务 II. 对于朋友推荐的任务，我们的目标是为用户 v 推荐其潜在朋友的用户的排序列表。

我们选择这两个任务是因为它们分别代表移动轨迹挖掘和社交网络分析的两个方面，且在 LBSN 中被广泛研究。

4.3.2　联合模型

在这一小节中，我们提出了一种用于生成社交网络和移动轨迹数据的神经网络模型。下面，我们首先研究如何建模每个单独的部分。然后，我们将介绍联合模型和参数学习算法。在介绍模型细节之前，我们首先在表4.1中总结了本章中使用的符号。

4.3.3　建模社交网络的构建

最近，网络表示学习受到了广泛研究[1,2,29,38]，它提供了一种使用低维嵌入向量探索网络结构模式的方法。研究者们已经证明在许多与网络不直接相关的任务（例如统计预测[77]）中，网络表示作为重要特征是十分有效的。在我们的任务中，我们基于两方面因素来刻画网络表示。首先，用户可能与他们的朋友具有类似的访问行为，我们可以利用用户链接来探索类似的访问模式。其次，网络结构可以被用作辅助信息以增强轨迹建模。

我们用 d 维嵌入向量 $F_v \in \mathbb{R}^d$ 来代表用户 v 的网络表示，矩阵 $F \in \mathbb{R}^{|V| \times d}$ 代表所有节点的网络表示。我们利用社交网络上的用户链接来学习网络表示，并对用户的结构模式的信息进行编码。

社交网络将基于用户的网络表示 F 构建。我们首先研究如何建模一条边 $v_i \rightarrow$

表 4.1　本章使用的主要符号

符号	描述
V, E	节点和边集合
L	位置集合
T_v, T_v^j	用户 v 的轨迹和第 j 个子轨迹
m_v	用户 v 的轨迹 T_v 中的子轨迹个数
$m_{v,j}$	用户 v 的轨迹 T_v 中的第 j 个子轨迹中的位置个数
$l_i^{(v,j)}$	用户 v 的第 j 个子轨迹的第 i 个位置
U_{l_i}	建模部分位置 l_i 的表示
U'_{l_i}	预测部分位置 l_i 的表示
P_v, F_v	用户 v 的兴趣偏好和好友表示
F'_v	用户 v 的好友上下文表示
S_i	访问位置 l_{i-1} 后的短期上下文表示
h_t	访问位置 l_{i-1} 后的长期上下文表示

v_j 的生成概率，即 $\Pr[(v_i, v_j) \in E]$。主要思路是，如果两个用户 v_i 和 v_j 在网络上形成了好友链接，它们的网络表示应该是相似的。换句话说，在相应的两个网络表示之间的内积 $F_{v_i}^\top \cdot F_{v_j}$ 将为两个连接的用户给出较大的相似度。一个潜在的问题是这样的建模只能处理无向网络。为了同时处理无向和有向网络，我们提出为用户 v_j 引入上下文表示 F'_{v_j}。给定有向边 $v_i \to v_j$，我们将表示相似度建模为 $F_{v_i}^\top \cdot F'_{v_j}$ 以取代 $F_{v_i}^\top \cdot F_{v_j}$。上下文表示仅用于网络构建部分，我们将边 $v_i \to v_j$ 的概率定义为

$$\Pr[(v_i, v_j) \in E] = \sigma(-F_{v_i}^\top \cdot F'_{v_j}) = \frac{1}{1 + \exp(-F_{v_i}^\top \cdot F'_{v_j})}. \tag{4-1}$$

当处理无向图时，好友对 (v_i, v_j) 将会被分为两条有向边 $v_i \to v_j$ 和 $v_j \to v_i$。对于边集 E 中不存在的边，我们使用下述公式

$$\Pr[(v_i, v_j) \notin E] = 1 - \sigma(-F_{v_i}^\top \cdot F'_{v_j}) = \frac{\exp(-F_{v_i}^\top \cdot F'_{v_j})}{1 + \exp(-F_{v_i}^\top \cdot F'_{v_j})}. \tag{4-2}$$

结合公式（4-1）和（4-2），我们实际上使用 Bernouli 分布对网络中的链接进行了建模。参考网络表示学习相关的研究[1]，我们假设每个用户对在生成过程中是独立的。也就是说不同 (v_i, v_j) 对的概率 $\Pr[(v_i, v_j) \in E|F]$ 是独立的。通过这个假

设，我们可以通过用户对来分解生成概率

$$\begin{aligned}\mathcal{L}(G) &= \sum_{(v_i,v_j)\in E} \log \Pr[(v_i,v_j)\in E] + \sum_{(v_i,v_j)\notin E} \log \Pr[(v_i,v_j)\notin E] \\ &= -\sum_{v_i,v_j} \log(1+\exp(-F_{v_i}^{\top}\cdot F'_{v_j})) - \sum_{(v_i,v_j)\notin E} F_{v_i}^{\top}\cdot F'_{v_j}.\end{aligned} \tag{4-3}$$

4.3.4　移动轨迹生成建模

本工作中，用户轨迹被形式化为有序的位置序列。因此，我们使用循环神经网络方法对轨迹生成过程进行建模。为了生成轨迹序列，我们基于四个重要因素逐一生成每个位置。我们首先总结以下四个因素：

- *一般访问偏好*: 用户的偏好或习惯直接决定了其访问行为。
- *好友影响*: 用户的访问行为很可能受到其朋友的影响。前人工作[60,68] 表明社交相关的用户倾向于访问相同的位置。
- *短期序列上下文*: 下一个位置与用户访问的最后几个位置密切相关。直观上，因为用户的访问行为通常与短时间内的单个活动或一系列活动相关，使得访问位置具有强相关性。
- *长期序列上下文*: 用户可能在长时间内对访问过的位置存在长期依赖性。长期依赖的一个具体情形是定期访问行为。例如，用户经常在每个夏天的假期中进行旅行。

前两个因素主要与用户和位置之间的交互有关。而后两个因素主要反映用户访问的位置之间的序列相关性。

4.3.4.1　一般访问偏好的建模

我们首先通过兴趣表示来描述一般访问偏好。我们用 d 维向量 $P_v \in \mathbb{R}^d$ 来代表用户 v 的访问偏好表示，矩阵 $P \in \mathbb{R}^{|V|\times d}$ 代表所有用户的访问偏好。访问偏好表示根据用户的访问行为，面向用户对位置集合的一般偏好进行编码。

我们假设一个人的一般访问兴趣相对稳定，并且在一段时期内变化不大。这种假设是合理的，因为用户通常具有固定的生活方式 (例如相对固定的居住区域)，其访问行为很可能会显示出一些全局模式。访问偏好表示旨在通过一个 d 维嵌入向量来捕捉和编码这种访问模式。

4.3.4.2　好友影响的建模

为了建模朋友的影响，一种直接的方法是使用一些正则化项来刻画来自两个连接用户的兴趣表示之间的相关性。然而，这种方法通常具有较高的计算复杂度。在本章中，我们采用了更灵活的方法，即将网络表示结合到轨迹生成过程中。因为网络表示是通过网络链接学习的，所以来自其朋友的信息可以隐含地被编码和使用。具体地，我们使用4.3.3节中介绍的网络表示 F_v。

4.3.4.3　短期序列上下文的建模

通常，用户在短时间内访问的位置密切相关。访问位置的短序列往往与某些活动有关。举例来说，‘‘家 → 交通 → 办公室’’ 对应从家到办公室的交通活动。此外，地理或交通限制在轨迹生成过程中起着重要作用。例如，用户更可能访问附近的位置。因此，当用户决定接下来要访问的位置时，其访问的最后几个位置对于下一位置预测是非常重要的。

基于上述考虑，我们将短时间内的最后几个访问位置视为序列历史，并基于它们预测下一个位置。为了捕捉短期访问依赖性，我们使用循环神经网络（RNN）来建模序列数据，并开发我们的模型。如图4.2所示，给定用户 v 轨迹中的第 j 个子序列 $T_v^j = \{l_1^{(v,j)}, l_2^{(v,j)} \dots l_{m_{v,j}}^{(v,j)}\}$，我们递归地定义短期序列相关性如下：

$$S_i = \tanh(U_{l_{i-1}} + W \cdot S_{i-1}), \tag{4-4}$$

其中 $S_i \in \mathbb{R}^d$ 是访问位置 l_{i-1} 后的状态的嵌入表示，$U_{l_i} \in \mathbb{R}^d$ 是位置 $l_i^{(v,j)}$ 的表示，$W \in \mathbb{R}^{d \times d}$ 是转移矩阵。RNN 类似于隐马尔可夫模型，因为序列相关性也通过两个连续状态之间的转换反映出来。一个主要的区别是，在 RNN 中，每个隐藏状态的特征是 d 维的嵌入向量。初始状态 S_0 对于所有用户都是一样的，因为短期相关性应该与我们模型中的用户偏好无关。公式（4-4）本质上是一个没有输出的 RNN 模型。每个状态对应的嵌入向量可以被理解为序列中的对应位置的信息概括。特别地，对应于最后位置的状态可以被认为是整个序列的嵌入表示。

4.3.4.4　长期序列上下文的建模

在上文中，短期连续上下文（我们的数据集中，平均有五个位置）旨在捕获短时间窗口中的序列相关性。在建模轨迹序列时，长期连续上下文也很重要。例如，用户可能会有一些定期访问模式。为了刻画长期依赖性，一种直接的方法是将

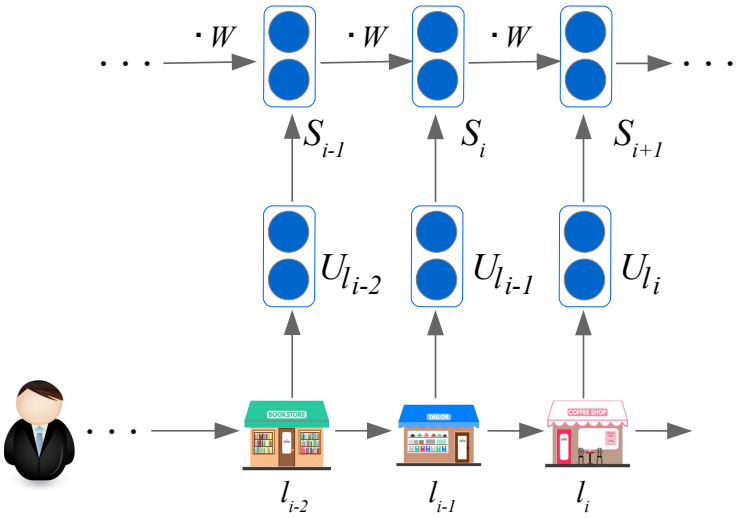


图 4.2　对短期序列上下文建模的循环神经网络的示意图

另一个 RNN 模型用于整个轨迹序列。然而，用户在长时间内产生的轨迹序列往往包含上百个甚至更多的地点位置。在长序列上使用 RNN 模型通常会遇到“梯度消失”的问题。

为了解决这个问题，我们采用带门循环单元（GRU）来捕捉轨迹序列中的长期依赖性。与传统的 RNN 相比，GRU 结合了几个额外的门单元来控制输入和输出。具体来说，我们在模型中使用了两个门：输入门和遗忘门。在输入和遗忘门的帮助下，GRU 的状态 C_t 即使在序列很长的情况下也能记住重要的内容，并在必要时忘记不太重要的信息。我们在图4.3展示了 GRU 的示意图。

形式化地，考虑位置序列 $\{l_1, l_2, \ldots, l_m\}$，我们将初始状态表示为 $C_0 \in \mathbb{R}^d$ 和 $h_0 = \tanh(C_0) \in \mathbb{R}^d$。在第 t 步，新的备选状态按下式更新

$$\widetilde{C}_t = \tanh(W_{c_1} U_{l_t} + W_{c_2} h_{t-1} + b_c), \tag{4-5}$$

其中 $W_{c_1} \in \mathbb{R}^{d \times d}$ 和 $W_{c_2} \in \mathbb{R}^{d \times d}$ 是模型参数，U_{l_t} 是位置 l_t 的表示，和短期序列建模中的参数相同，h_{t-1} 是最后一步的嵌入表示，$b_c \in \mathbb{R}^d$ 是偏置向量。注意 $\widetilde{C}_t$ 的计算和 RNN 一样。

GRU 并不像 RNN 一样直接将 $\widetilde{C}_t$ 用作隐状态。取而代之的是，GRU 在最后的

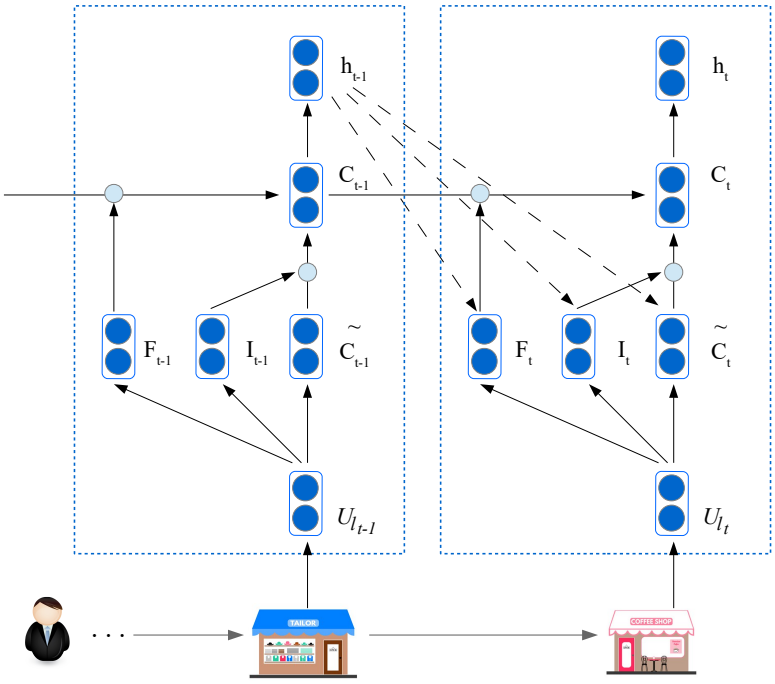


图 4.3　GRU 架构的示意图。$\widetilde{C}_t$ 是备选状态，当前状态 C_t 是上个状态 C_{t-1} 和当前备选状态 $\widetilde{C}_t$ 的混合。I_t 和 F_t 分别是控制混合比例的输入和遗忘门。

状态 C_{t-1} 和新的备选状态 $\widetilde{C}_t$ 之间找到了平衡：

$$C_t = i_t * \widetilde{C}_t + f_t * C_{t-1}, \tag{4-6}$$

其中 $*$ 是按位乘操作，$i_t, f_t \in \mathbb{R}^d$ 分别是输入、遗忘门。

输入、遗忘门 $i_t, f_t \in \mathbb{R}^d$ 定义为

$$i_t = \sigma(W_{i_1} U_{l_t} + W_{i_2} h_{t-1} + b_i), \tag{4-7}$$

和

$$f_t = \sigma(W_{f_1} U_{l_t} + W_{f_2} h_{t-1} + b_f), \tag{4-8}$$

其中 $\sigma(\cdot)$ 是 sigmoid 函数，$W_{i_1}, W_{i_2} \in \mathbb{R}^{d\times d}$ 和 $W_{f_1}, W_{f_2} \in \mathbb{R}^{d\times d}$ 是输入、遗忘门的参数，$b_i, b_f \in \mathbb{R}^d$ 是偏置向量。

最后，第 t 步的长期序列表示如下

$$h_t = \tanh(C_t). \tag{4-9}$$

和公式（4-4）类似，h_t 编码了轨迹中前 t 个位置的信息。我们可以在每访问一个位置后递归地计算该表示。

4.3.4.5　生成轨迹数据的最终目标函数

基于上述讨论，我们现在提出用于生成轨迹数据的目标函数。给定用户 v 的轨迹序列 $T_v = \{l_1^{(v)}, l_2^{(v)}, \ldots, l_m^{(v)}\}$，我们根据链式法则将对数似然分解如下

$$\begin{aligned}\mathcal{L}(T_v) &= \log \Pr[l_1^{(v)}, l_2^{(v)}, \ldots, l_m^{(v)} | v, \Phi] \\ &= \sum_{i=1}^{m} \log \Pr[l_i^{(v)} | l_1^{(v)}, \ldots, l_{i-1}^{(v)}, v, \Phi],\end{aligned} \tag{4-10}$$

其中 Φ 表示所有相关参数。$\mathcal{L}(T_v)$ 是基于用户 v 和参数 Φ 的对数概率的和。注意 T_v 被分为 m_v 个子轨迹 $T_v^1, \ldots, T_v^{m_v}$。令 $l_i^{(v,j)}$ 为第 j 个子轨迹中的第 i 个位置。$l_i^{(v,j)}$ 的上下文位置包括同一个子轨迹的前 $i-1$ 个位置 (即 $l_1^{(v,j)} \ldots l_{i-1}^{(v,j)}$)，记为 $l_1^{(v,j)} : l_{i-1}^{(v,j)}$，前 $j-1$ 个子轨迹中的所有位置 (即 $T_v^1, \ldots, T_v^{j-1}$) 记为 $T_v^1 : T_v^{j-1}$。基于这些符号，我们重写公式（4-10）如下

$$\mathcal{L}(T_v) = \sum_{i=1}^{m} \log \Pr[l_i^{(v,j)} | \underbrace{l_1^{(v,j)} : l_{i-1}^{(v,j)}}_{\text{短期上下文}}, \underbrace{T_v^1 : T_v^{j-1}}_{\text{长期上下文}}, v, \Phi]. \tag{4-11}$$

给定目标位置 $l_i^{(v,j)}$，$l_1^{(v,j)} : l_{i-1}^{(v,j)}$ 对应短期上下文，$T_v^1 : T_v^{j-1}$ 对应长期上下文，v 对应用户上下文。关键问题变为如何建模条件概率 $\Pr[l_i^{(v,j)} | l_1^{(v,j)} : l_{i-1}^{(v,j)}, T_v^1 : T_v^{j-1}, v, \Phi]$。

对于短期上下文，我们使用公式（4-4）中介绍的 RNN 模型来刻画位置序列 $l_1^{(v,j)} : l_{i-1}^{(v,j)}$。我们用 S_i^j 代表访问了第 j 个子轨迹中的第 i 个位置后的短期上下文表示；对于长期上下文，前面子轨迹 $T_v^1 \ldots T_v^{j-1}$ 中的位置由公式（4-6）中的 GRU 建模。我们用 h^j 代表访问了前 j 个子轨迹中的位置后的长期上下文表示。我们在图4.4中展示了结合短期与长期上下文的模型示意图。

到目前为止，给定目标位置 $l_i^{(v,j)}$，我们已经得到了四个因素的表示：网络表示 (F_v)，用户偏好表示 (P_v)，短期上下文表示 S_{i-1}^j 和长期上下文表示 h^{j-1}，我们将它

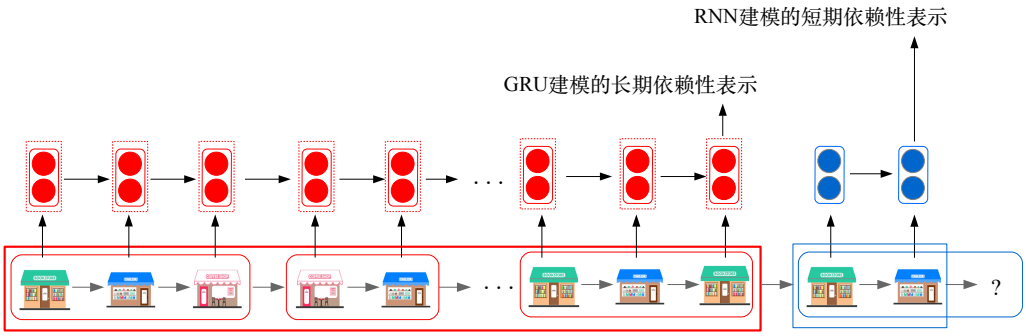


图 4.4　结合短期与长期序列上下文的模型示意图。圆角矩形内的位置是一个子轨迹。红色和蓝色矩形内的位置分别为长期和短期序列上下文。“?” 是需要预测的下一个位置。

们拼接成一个向量 $R_v^{(i,j)} = [F_v; P_v; S_{i-1}^j; h^{j-1}] \in \mathbb{R}^{4d}$ 并用于下一个位置预测。给定表示 $R_v^{(i,j)}$，我们定义 $l_i^{(v,j)}$ 的概率为

$$\begin{aligned} & \Pr[l_i^{(v,j)} | l_1^{(v,j)} : l_{i-1}^{(v,j)}, T_v^1 : T_v^{j-1}, v, \Phi] \\ = \; & \Pr[l_i^{(v,j)} | R_v^{(i,j)}] \\ = \; & \frac{\exp(R_v^{(i,j)} \cdot U'_{l_i^{(v,j)}})}{\sum_{l \in L} \exp(R_v^{(i,j)} \cdot U'_l)}, \end{aligned} \tag{4-12}$$

其中参数 $U'_l \in \mathbb{R}^{4d}$ 是位置 $l \in L$ 用于预测的表示。注意这个表示 U'_l 和长短期上下文建模中的位置表示 $U_l \in \mathbb{R}^d$ 完全不同。我们可以通过将所有位置的对数概率相加来计算轨迹生成的总体对数似然。

4.3.5　整体模型

我们的整体模型是两个部分的目标函数之间的线性组合。给定社交关系网络 $G = (V, E)$ 和用户轨迹 T，我们有下面的对数似然函数

$$\begin{aligned} \mathcal{L}(G, T) &= \mathcal{L}_{\text{network}}(G) + \mathcal{L}_{\text{trajectory}}(T) \\ &= \mathcal{L}(G) + \sum_{v \in V} \mathcal{L}(T_v). \end{aligned} \tag{4-13}$$

其中 $\mathcal{L}_{\text{network}}(G)$ 由公式（4-3）定义，$\mathcal{L}_{\text{trajectory}}(T) = \sum_{v \in V} \mathcal{L}(T_v)$ 由公式（4-11）定义。我们将模型命名为 *Joint Network and Trajectory Model* (***JNTM***)。

我们在图4.5中展示了整体模型 JNTM 的示意图。我们的模型是一个用于生成社交网络和用户轨迹的三层神经网络。在训练中，我们要求提供社交网络和用户轨迹作为训练模型的目标输出。基于这些数据，我们的模型包含两个目标函数。为了生成社交网络，模型引入了基于网络的用户表示；为了生成用户轨迹，模型考虑了四个因素：基于网络的表示，一般访问偏好，短期和长期序列上下文。这两部分通过共享基于网络的用户表示来联系在一起。

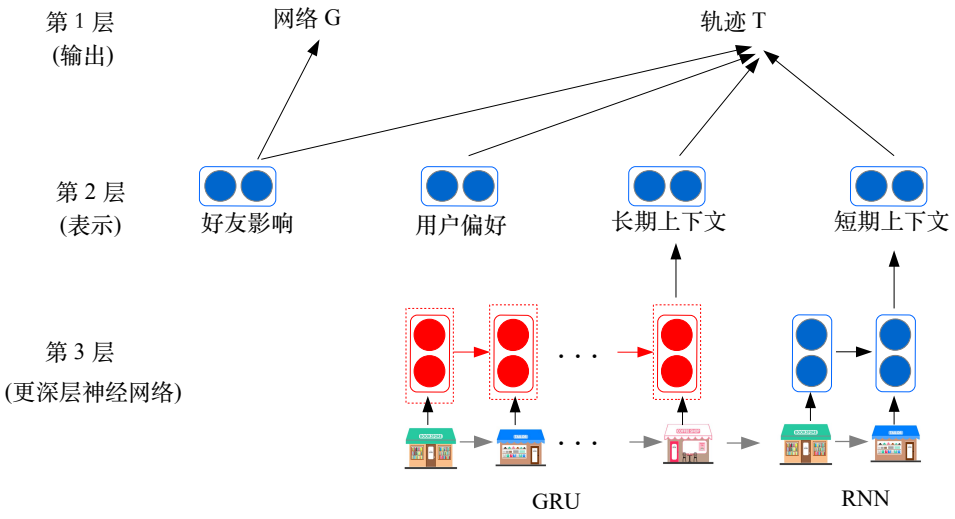


图 4.5　整体模型 JNTM 的示意图

4.3.6　参数学习

现在我们将介绍如何训练我们的模型并学习参数，包括用户偏好表示 $P \in \mathbb{R}^{|V|\times d}$，用户好友网络表示 $F, F' \in \mathbb{R}^{|V|\times d}$，位置表示 $U \in \mathbb{R}^{|L|\times d}, U' \in \mathbb{R}^{|L|\times 4d}$，初始短序列表示 $S_0 \in \mathbb{R}^d$，转移矩阵 $W \in \mathbb{R}^{d\times d}$，初始 GRU 状态 $C_0 \in \mathbb{R}^d$ 和 GRU 参数 $W_{i_1}, W_{i_2}, W_{f_1}, W_{f_2}, W_{c_1}, W_{c_2} \in \mathbb{R}^{d\times d}, b_i, b_f, b_c \in \mathbb{R}^d$ 。

负采样方法。公式（4-3）中的网络生成的对数似然包括 $|V| \times |V|$ 项，需要至少 $O(|V|^2)$ 的时间复杂度，非常耗时。因此我们使用 NLP 领域经常使用的负采样方法[3] 加速我们的训练过程。

注意真实网络数据一般是稀疏的，也就是 $O(E) = O(V)$。相连的节点对数量（正例）远远少于不相连节点对的数量（负例）。负采样的核心思想是大多数节点对都是负例，而我们不需要计算所有节点对。取而代之的是，我们计算所有的相连

节点对和 n_1 随机不相连节点对作为近似，其中 $n_1 \ll |V|^2$ 是负例数。在我们的实验中，我们设 $n_1 = 100|V|$。对数似然可以被写为

$$\mathcal{L}(G|F, F') = \sum_{(v_i, v_j) \in E} \log \Pr[(v_i, v_j) \in E] + \sum_{k=1, (v_{ik}, v_{jk}) \notin E}^{n_1} \log \Pr[(v_{ik}, v_{jk}) \notin E]. \quad (4\text{-}14)$$

网络生成部分的似然计算只包括 $O(E + n_1) = O(V)$ 项。

另一方面，公式（4-12）的计算需要至少 $O(|L|)$ 时间因为分母包含 $|L|$ 项。注意，我们需要对每个位置进行该条件概率的计算。因此轨迹生成的计算需要至少 $O(|L|^2)$ 时间，效率很低。类似地，我们不计算所有的分母项。我们只计算位置 $l_i^{(v,j)}$ 和其他 n_2 个随机位置。在本章工作中，我们设置 $n_2 = 100$。我们重写公式（4-12）为

$$\Pr[l_i^{(v,j)} | R_v^{(i,j)}] = \frac{\exp(R_v^{(i,j)} \cdot U'_{l_i^{(v,j)}})}{\exp(R_v^{(i,j)} \cdot U'_{l_i^{(v,j)}}) + \sum_{k=1, l_k \neq l_i^{(v,j)}}^{n_2} \exp(R_v^{(i,j)} \cdot U'_{l_k})}. \quad (4\text{-}15)$$

分母的计算将只包含 $O(n_2 + 1) = O(1)$ 项。

我们通过 back propagation through time (BPTT)[96] 来计算参数的梯度。之后，参数可以通过 AdaGrad[97] 更新。

复杂度分析：我们首先给出了时间成本的复杂度分析。用户 v 的网络生成需要 $O(d)$ 时间来计算对数似然和 F_v 以及 F' 中对应行的梯度。所以网络部分的复杂度是 $O(d|V|)$。在轨迹生成中，我们将所有的位置访问记录记为 $|D|$。GRU 的正向和反向传播需要 $O(d^2|D|)$ 时间。RNN 的每步需要 $O(d^2)$ 时间来更新状态表示并计算 S_0, U, W 的梯度。对数似然和 $U', F_v, P_v, S_{i-1}^j, h^{j-1}$ 的梯度计算需要 $O(d^2)$ 时间。所以我们模型的整体复杂度是 $O(d^2|D| + d|V|)$。注意，表示维度 d 和每个用户/位置的负样本数远小于数据规模大小 $|V|$ 和 $|D|$。所以我们的算法 JNTM 的时间复杂度与数据大小呈线性关系，且适用于大型数据集。虽然我们模型的训练时间复杂度相对较高，但测试时间复杂度很小。在测试阶段向用户做位置推荐时，只需要 $O(d)$ 时间来更新 RNN/GRU 的状态，$O(d)$ 时间计算单个位置的评分。维度 d 一般比较小，意味着我们的算法对于在线推荐十分高效。

就空间复杂度而言，网络表示 F 和位置表示 U 一共占用了 $O((|V| + |L|)d)$ 的空间。其他参数的空间占用只有 $O(d^2)$，可以几乎忽略。因此，我们的模型的空间复杂度和以前的模型相当，例如 FPMC[93]，PRME[94] 和 HRM[95]。

4.4　实验结果

在本节中，我们将评估我们提出的模型 JNTM 的性能。我们考虑了下一个位置推荐和好友推荐两个应用任务。下面我们将介绍数据收集，基线方法，参数设置和评估指标。然后我们将展示实验结果与相关分析。

4.4.1　数据收集

我们使用了两个公开可用的 LBSN 数据集①[48]，即 Gowalla 和 Brightkite。Gowalla 和 Brightkite 为用户提供了移动应用程序。例如，使用 Brightkite，你可以使用手机内置的 GPS 跟踪你的朋友或附近的任何其他 BrightKite 用户; Gowalla 也具有类似的功能：使用 GPS 数据显示你以及你附近用户的位置。

这两个数据集提供了用户链接和用户位置访问记录。用户链接表示好友关系，位置访问记录包含位置 ID 和相应的时间戳。我们将访问记录信息转为轨迹序列。和前人工作[65] 类似，我们在两次连续访问之间的间隔大于六小时的地方切分轨迹。我们在两个数据集上执行了一些预处理步骤。对 *Gowalla*，我们去掉了所有少于 10 个位置访问的用户和少于 15 个被访问的位置，最终得到了 837, 352 条子轨迹。对 *Brightkite*，因为这个数据较小，我们去掉了所有少于 10 个位置访问的用户和少于 5 个被访问的位置，最终得到了 503, 037 条子轨迹。表4.2展示了数据集预处理后的统计信息。我们使用的数据集比前人工作[65,94] 拥有更大的规模。

表 4.2　数据集统计。$|V|$: 节点数; $|E|$: 边数; $|D|$: 访问记录数; $|L|$: 位置数.

Dataset	$\|V\|$	$\|E\|$	$\|D\|$	$\|L\|$
Gowalla	37,800	390,902	2,212,652	58,410
Brightkite	11,498	140,372	1,029,959	51,866

我们的主要假设是社交链接和移动轨迹行为之间存在密切关联。为了验证这一假设，我们构建了一个实验来揭示这两个因素之间的相关模式。对于每个用户，其访问过的位置构成一个位置集合。然后我们可以使用重合系数来测量两个用户的位置集合之间的相似度②。对于随机好友用户对，在 Brightkite 和 Gowalla 上的用户间平均重合系数分别为 11.1% 和 15.7%。作为参照，对于随机非好友用户对，在 Brightkite 和 Gowalla 上的用户间平均重合系数都降到了 0.5%。这一发现表明有社交连接的用户确实拥有更相似的位置访问特征。接下来我们将验证具有相似轨迹行为的两个用户是否更有可能有社交链接。我们发现 Brightkite 和 Gowalla 数据上

① http://snap.stanford.edu/data/

② https://en.wikipedia.org/wiki/Overlap_coefficient

两个随机用户是好友的概率分别为 0.1% 和 0.03%。但如果我们随机选取两个至少访问过 3 个相同位置的用户，他们是好友的概率将增长到 9% 和 2%。上述两个观察结果表明，社交关系与 LBSN 中的移动轨迹行为密切相关。

4.4.2　评估任务与基线方法

4.4.2.1　下一个位置推荐

对于下一个位置推荐任务，我们考虑了以下基线方法：

Paragraph Vector (PV)[98] 是使用简单神经网络架构的句子和文档的表示学习模型。为了建模轨迹数据，我们将每个位置视为一个单词，将每个用户视为一个拥有多个单词的段落。

Feature-Based Classification (FBC) 通过将其看作多类分类问题来解决下一个位置推荐任务。用户特征和位置特征分别通过 DeepWalk 算法[1] 和 word2vec[3] 学习得到。之后这些特征被送入 softmax 分类器用于预测。

FPMC[93] 是一种最先进的推荐算法。FPMC 通过计算基于马尔可夫链假设的转移概率，来对所有用户的转移矩阵的张量进行分解并预测下一个位置。虽然 FPMC 最初是针对于产品推荐而提出的，但我们可以很简单地让 FPMC 应用于下一个位置推荐任务。

PRME[94] 通过在不同的向量空间中对用户-位置对和位置-位置对进行建模来扩展 FPMC。PRME 在下一个位置推荐任务中拥有目前最好的效果。

HRM[95] 是下个交易篮推荐的最新算法。通过将每个子轨迹作为交易篮，我们可以轻松地将 HRM 用于下一个位置推荐。这是第一个将分布式表示学习应用于推荐问题的研究。

我们选择这五个基线，因为它们代表了不同的推荐算法。PV 基于简单神经网络，FBC 是使用嵌入特征的传统分类模型，FPMC 主要基于矩阵分解框架，PRME 改进了 FPMC 使其适用于下一个位置推荐，HRM 使用了分布式表示学习来进行下一个交易篮推荐。

下面我们将数据划分为训练集和测试集。一个用户的前 90% 子轨迹被划入训练集，余下的 10% 划为测试集。为了调参，我们将训练数据的后 10% 作为验证集。

给定一个用户，我们按照顺序，逐个地预测测试集中的位置。对于每个需要被预测的位置，我们为用户推荐五或十个位置。对 JNTM，我们按公式（4-12）中的对数似然对位置进行排序。对于基线方法，FPMC 和 HRM 按位置的转移概率排序，PRME 按转移距离排序。PV 和 FBC 的预测结果可以由对应算法 softmax 层的

输出得到。我们将 Recall@5 和 Recall@10 作为评价指标，其中 Recall@K 定义为

$$Recall@K = \frac{K \text{ 个推荐位置中正确的位置数}}{\text{测试集中的总位置数}}.$$

注意我们也可以使用另一常用指标 Precision@K。在实验中，这两个指标实际上是正相关的。即如果方法 A 比方法 B 有更高的 Recall@K 值，那么 A 也会有比 B 更高的 Precision@K。所以我们省略了 Precision@K 的结果。

4.4.2.2　好友推荐

对于好友推荐的任务，我们根据使用的数据考虑了三种类型的基线，包括只使用网络数据的基线 (即 DeepWalk)，只使用轨迹数据的基线 (即 PMF) 和同时使用网络和轨迹的基线 (即 PTE 和 TADW)。

DeepWalk[1] 是一种先进的网络表示学习方法，它从随机游走序列中学习节点嵌入。它首先生成随机游走路径，并应用词嵌入技术来学习网络节点的表示。

PMF[99] 是一种基于用户-物品矩阵分解的通用协同过滤方法。在我们的实验中，我们使用轨迹数据构建用户-位置矩阵，然后我们利用用户表示来进行好友推荐。

PTE[20] 是半监督文本嵌入学习算法。我们去掉了监督部分，并优化邻接矩阵和用户-位置共现矩阵，使其适用于无监督嵌入学习。PTE 建模了条件概率 $p(v_j|v_i)$ 来表示 v_i 和 v_j 相连接的概率。我们计算该条件概率用于好友推荐。

TADW[28] 进一步扩展了 DeepWalk 来利用网络中的文本信息。我们可以通过忽略位置的顺序信息，用用户-位置共现矩阵来替换 TADW 中的文本特征矩阵。TADW 定义了一个邻近度矩阵，其中矩阵的每个元素表示相应用户之间关系的强度。我们使用邻近度矩阵的相应元素来排序候选用户并进行推荐。

我们随机抽取了 20% ~ 50% 的好友链接作为训练集，其余的作为测试。我们为每个用户推荐 5 或 10 个好友，并报告 Recall@5 和 Recall@10。具体地，对每个用户 v，我们将训练集中不是其朋友的所有其他用户作为候选用户。然后，我们对候选用户进行排序，并推荐分数最高的 5 或 10 个。当我们为用户 v_i 推荐朋友时，需要计算用户 v_j 的排序分数，DeepWalk 和 PMF 使用它们用户表示之间的余弦相似度，PTE 使用条件概率 $p(v_j|v_i)$，TADW 使用邻近度矩阵 A 中的对应单元 A_{ij}。对于我们的模型，我们使用公式（4-1）中的对数似然。

基线方法和我们的模型都涉及一个重要的参数，即表示维度。我们在 25 和 100 间进行网格搜索，并使用验证集中的最优值。基线或我们的模型中的其他参数可

以以类似的方式进行调整。对于我们的模型，学习率和负样本数根据经验分别设置为 0.1 和 100。我们根据均匀分布 $U(-0.02, 0.02)$ 随机初始化参数。

所有实验都在 12 核 CPU 服务器上执行，CPU 类型为 Intel Xeon E5-2620 @ 2.0GHz。

4.4.3　下一个位置推荐任务实验结果

表4.3展示了不同方法在下一个位置推荐任务上的实验结果。与 FPMC 和 PRME 相比，HRM 模拟了连续子轨迹之间的序列相关性，而忽略了子轨迹内部的序列相关性。Brightkite 数据集中，子轨迹中的平均位置数要远远小于 Gowalla 中的。因此，短期序列上下文对 Gowalla 比对于 Brightkite 更加有用。表4.3中的实验结果验证了我们的直觉：HRM 在 Brightkite 数据上比 FPMC and PRME 表现要好；而 PRME 在 Gowalla 数据上表现最好。

如表4.3所示，我们的模型 JNTM 一致地超过其他基线方法。与 Brightkite 数据集上的最佳基线 HRM 和 Gowalla 数据集上的最佳基线 FBC 进行比较，JNTM 在 Recall@5 评分上有 4.9% 和 4.4% 的提升。注意 JNTM 考虑了四个因素，包括用户偏好，好友影响，短期和长期的序列上下文。所有基线方法仅考虑了用户偏好（或好友关系）和单一类型的序列上下文。因此，JNTM 在两个数据集上都达到了最好的效果。

表 4.3　下一个位置推荐任务上不同方法的实验结果

数据集	Brightkite			Gowalla		
评价指标 (%)	R@1	R@5	R@10	R@1	R@5	R@10
PV	18.5	44.3	53.2	9.9	27.8	36.3
FBC	16.7	44.1	54.2	13.3	34.4	42.3
FPMC	20.6	45.6	53.8	10.1	24.9	31.6
PRME	15.4	44.6	53.0	12.2	31.9	38.2
HRM	17.4	46.2	56.4	7.4	26.2	37.0
JNTM	**22.1**	**51.1**	**60.3**	**15.4**	**38.8**	**48.1**

上述结果是基于对所有用户的结果进行平均而得到。在推荐系统中，一个重要的问题是方法在冷启动设置中的表现如何，即新用户或新物品。为了检验对于仅拥有很少位置访问信息的新用户的有效性，我们在表4.4中展示了对于拥有少于五条子轨迹的用户的 Recall@5。冷启动场景中，一个常用手段是加入辅助信息 (例如用户链接[60] 和文本信息[87–89]) 来缓解数据稀疏性。对于我们的模型，我们可以利用网络数据学习得到的用户表示在一定程度上改善新用户的推荐性能。实际上，

网络表示已应用于多个与网络不直接相关的任务，如职业推荐[100] 和文本分类[28]。实验结果表明我们的模型 JNTM 有一定在冷启动设置中完成下一个位置推荐的能力。

表 4.4　对于拥有不多于五条子轨迹的用户的下一个位置推荐实验结果

数据集	Brightkite			Gowalla		
评价指标 (%)	R@1	R@5	R@10	R@1	R@5	R@10
PV	13.2	22.0	26.1	4.6	7.8	9.2
FBC	9.0	29.6	39.5	4.9	12.0	16.3
FPMC	17.1	30.0	33.9	5.5	13.5	18.5
PRME	22.4	36.3	40.0	7.2	12.2	15.1
HRM	12.9	31.2	39.7	5.2	15.2	21.5
JNTM	**28.4**	**53.7**	**59.3**	**10.2**	**24.8**	**32.0**

注意上述实验基于普通的下一个位置推荐，我们不会检查用户之前是否访问过系统推荐的位置。为了进一步测试我们算法的有效性，我们对先前研究[94] 提出的下一个新位置推荐任务进行了实验。在这个设置下，我们只向用户推荐其没有访问过的新位置。特别地，我们在推荐时只对用户所有没访问过的位置进行排序[94]。我们将实验结果列在表4.5中。在两个数据集的下一个新位置推荐任务中，我们的方法始终优于所有基线。结合表4.3和4.4，我们可以看到，与这些基线相比，我们的模型 JNTM 在下一个位置推荐任务中更加有效。

表 4.5　下一个新位置推荐任务上不同方法的实验结果

数据集	Brightkite			Gowalla		
评价指标 (%)	R@1	R@5	R@10	R@1	R@5	R@10
PV	0.5	1.5	2.3	1.0	3.3	5.3
FBC	0.5	1.9	3.0	1.0	3.1	5.1
FPMC	0.8	2.7	4.3	2.0	6.2	9.9
PRME	0.3	1.1	1.9	0.6	2.0	3.3
HRM	1.2	3.5	5.2	1.7	5.3	8.2
JNTM	**1.3**	**3.7**	**5.5**	**2.7**	**8.1**	**12.1**

在上文中，我们已经展示了所提出的模型 JNTM 对下一个位置推荐任务的有效性。由于轨迹数据本身就是序列数据，JNTM 模型中的短期和长期序列上下文利用了循环神经网络的灵活性来建模。现在我们研究序列建模对当前任务的影响。

我们准备了 JNTM 的三个变体：

- JNTM$_{base}$：删除了短期和长期上下文，仅使用用户偏好表示和网络表示来生成轨迹数据。
- JNTM$_{base+long}$：在 JNTM$_{base}$ 基础上加入了长期上下文。
- JNTM$_{base+long+short}$：在 JNTM$_{base}$. 基础上同时加入了短期和长期上下文。

表 4.6　JNTM 的三种变体在下一个位置推荐任务上的性能比较

数据集	Brightkite			Gowalla		
评价指标 (%)	R@1	R@5	R@10	R@1	R@5	R@10
JNTM$_{base}$	20.2	49.3	59.2	12.6	36.6	45.5
JNTM$_{base+long}$	20.4 (+2%)	50.2 (+2%)	59.8 (+1%)	13.9 (+10%)	36.7 (+0%)	45.6 (+0%)
JNTM$_{base+long+short}$	**22.1**(+9%)	**51.1**(+4%)	**60.3**(+**2**%)	**15.4**(+**18**%)	**38.8**(+6%)	**48.1**(+6%)

表 4.7　JNTM 的三种变体在下一个新位置推荐任务上的性能比较

数据集	Brightkite			Gowalla		
评价指标 (%)	R@1	R@5	R@10	R@1	R@5	R@10
JNTM$_{base}$	0.8	2.5	3.9	0.9	3.3	5.5
JNTM$_{base+long}$	1.0 (+20%)	3.3 (+32%)	4.8 (+23%)	1.0 (+11%)	3.5 (+6%)	5.8 (+5%)
JNTM$_{base+long+short}$	**1.3**(+63%)	**3.7**(+48%)	**5.5**(+41%)	**2.7**(+200%)	**8.1**(+145%)	**12.1**(+120%)

表4.6和4.7展示了三个 JNTM 模型变体在 Brightkite 和 Gowalla 数据集上的实验结果。括号内的数字代表相比于 JNTM$_{base}$ 的相对提升。我们可以观察到性能排序：JNTM$_{base}$ < JNTM$_{base+long}$ < JNTM$_{base+long+short}$。观察结果表明，两种序列上下文对于提高下一个位置推荐的性能都很有用。在普通的下一个位置推荐中 (即新老位置都会被推荐)，我们可以看到来自短期和长期背上下文的提升并不显著。解释是用户可能会表现出重复地访问某个位置的行为，因此用户偏好对于推荐性能的影响比序列上下文更重要。而对于下一个新位置推荐，序列上下文尤其是短期上下文会产生比基线更大的提升。这些结果表明，在新位置推荐任务中，序列影响比用户偏好更重要。我们的发现也和前人工作[94] 一致，即序列上下文对于下一个新位置推荐非常重要。

4.4.4　好友推荐任务实验结果

我们继续介绍和分析好友推荐任务的实验结果。表4.8和4.9展示了当训练比例从 20% 变化到 50% 的实验结果。

在基线方法中，DeepWalk 表现最好，甚至超过了同时使用网络数据和轨迹数据的基线 (PTE 和 TADW)。一个主要原因是 DeepWalk 天然适用于网络链接的重

表 4.8　Brightkite 数据集上好友推荐任务实验结果

训练比例	20%		30%		40%		50%	
评价指标 (%)	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10
DeepWalk	2.3	3.8	3.9	6.7	5.5	9.2	7.4	12.3
PMF	2.1	3.6	2.1	3.7	2.3	3.4	2.3	3.8
PTE	1.5	2.5	3.8	4.7	4.0	6.6	5.1	8.3
TADW	2.2	3.4	3.6	3.9	2.9	4.3	3.2	4.5
JNTM	**3.7**	**6.0**	**5.4**	**8.7**	**6.7**	**11.1**	**8.4**	**13.9**

表 4.9　Gowalla 数据集上好友推荐任务实验结果

训练比例	20%		30%		40%		50%	
评价指标 (%)	R@5	R@10	R@5	R@10	R@5	R@10	R@5	R@10
DeepWalk	2.6	3.9	5.1	8.1	**7.9**	**12.1**	**10.5**	**15.8**
PMF	1.7	2.4	1.8	2.5	1.9	2.7	1.9	3.1
PTE	1.1	1.8	2.3	3.6	3.6	5.6	4.9	7.6
TADW	2.1	3.1	2.6	3.9	3.2	4.7	3.6	5.4
JNTM	**3.8**	**5.5**	**5.9**	**8.9**	**7.9**	11.9	10.0	15.1

建，并采用分布式表示方法来建模拓扑结构。尽管 PTE 和 TADW 同时利用了网络和轨迹数据，但它们的性能仍然很低。这两种方法也无法捕捉轨迹序列中的序列相关性。

我们的算法与最先进的网络嵌入方法 DeepWalk 相比效果相当，并且在网络结构信息稀疏时优于 DeepWalk。其解释是当网络信息不足时，轨迹信息更有用。随着网络信息变得稠密，轨迹信息不如用户链接信息更直接有用。为了验证这一解释，我们进一步报告了当训练比率为 50% 时，对于少于五个好友的用户的推荐结果。如表4.10所示，在 Brightkite 和 Gowalla 数据集上，我们的方法相比于 DeepWalk 分别有 2.1% 和 1.5% 的提升。结果表明，对于好友很少的用户，轨迹信息对于改善好友推荐的性能是有用的。

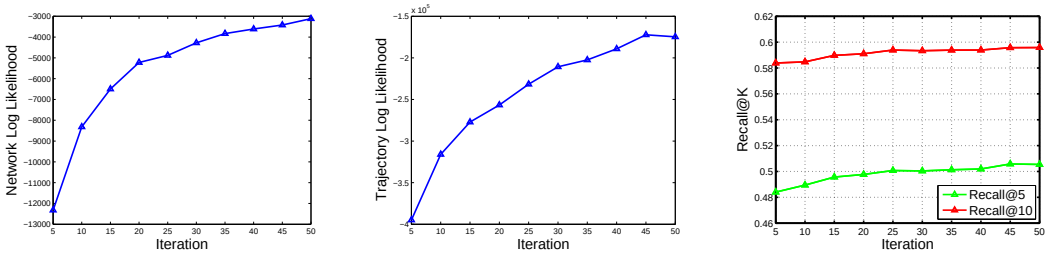
表 4.10　当训练比例为 50% 时，对于少于五个好友的用户的好友推荐实验结果

数据集	Brightkite		Gowalla	
评价指标 (%)	R@5	R@10	R@5	R@10
DeepWalk	14.0	18.6	19.8	23.5
JNTM	**16.1**	**20.4**	**21.3**	**25.5**

总结上述实验，我们的方法在下一个位置和好友推荐两个任务上明显优于现有的最先进方法。两项任务的实验结果证明了我们提出的模型的有效性。

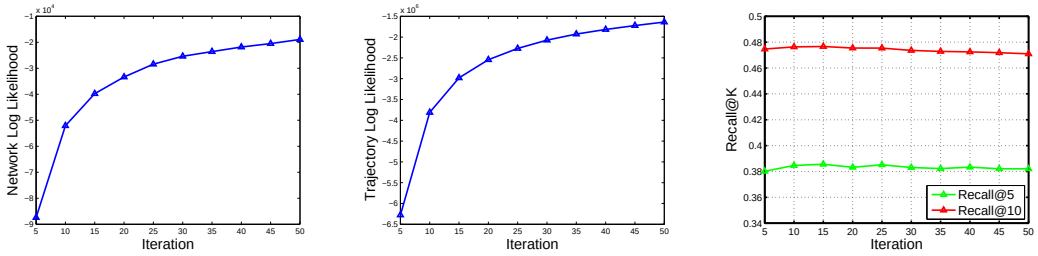
4.4.5　参数调试

在本小节中，我们将研究不同的参数设置将如何影响模型的性能。我们主要选择了两个重要参数，即迭代次数和嵌入维数。



(a) 网络对数似然　　(b) 轨迹对数似然　　(c) Recall@K

图 4.6　Brightkite 数据集上迭代次数参数的影响



(a) 网络对数似然　　(b) 轨迹对数似然　　(c) Recall@K

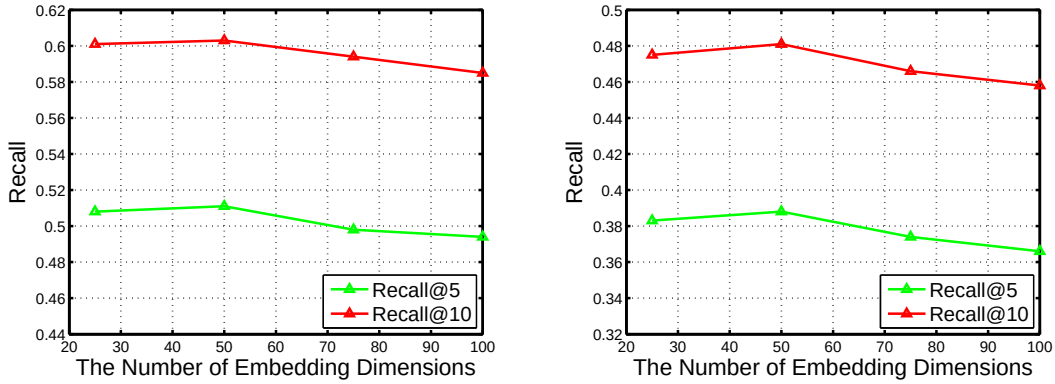
图 4.7　Gowalla 数据集上迭代次数参数的影响

我们让迭代次数从 5 增长到 50，并报告训练集上网络和轨迹数据的对数似然以及验证集上下一个位置推荐任务的 Recall@5 和 Recall@10 值。

图4.6和4.7展示了两个数据集上迭代次数参数的影响。可以看到我们的算法可以在 50 轮内收敛，对数似然的增长在 30 轮后开始变缓。另一方面，验证集上下一个位置推荐的性能相对稳定：JNTM 可以在 5 轮之后取得相对不错的效果。在 Brightkite 和 Gowalla 数据集上，召回率慢慢增长并在 45 轮和 15 轮分别达到最大。Gowalla 比 Brightkite 收敛的更快也更平滑，主要是因为 Gowalla 数据集有 3 倍多的位置访问信息，可以训练地更充分。然而，在到达召回率最高点之前，该模型可能已经过拟合了，因为召回率并不总是单调增长。另一个证据是，下一个新位置推荐的召回率在 10 轮后开始下降。为了避免过拟合问题，我们对 Brightkite 和 Gowalla 分别设置迭代轮数为 15 和 10。

嵌入表示的维度对我们模型的性能也至关重要。大的表示维数将具有更强的表达能力，但也可能导致过拟合。我们在下一个位置推荐任务上使用不同的嵌入

维度进行实验，并测量了它们在验证集上的效果。在图4.8中，当维度在 25 到 100 之间变化时，可以看到我们算法的性能相对稳定。当维度超过 50 时，召回率开始下降。我们最后设置维度为 50。



(a) Brightkite　　(b) Gowalla

图 4.8　不同维度参数的影响

4.4.6　计算效率

在这一部分，我们进行实验来检测我们的模型的时间和空间成本。我们在 Gowalla 数据集上进行实验，并选取基线方法 PRME[94] 作为比较。我们报告了两种方法的内存使用情况以用于空间复杂性分析和单个 CPU 上的运行时间以进行时间复杂度分析。由于两种方法具有相同的收敛迭代次数，因此我们报告每种算法的每次迭代的平均运行时间。训练和测试的运行时间将单独列出。神经网络模型的一个主要优点是它们可以通过支持的硬件进行加速 (如 GPU)。因此，我们还报告了使用 GPU 加速的模型变体的运行时间。特别地，我们使用 Tesla K40 GPU 和 TensorFlow[①]库训练模型。实验结果见表4.11。

表 4.11　内存占用 (MB) 和运行时间 (分钟)

模型	内存占用	训练时间	测试时间	训练时间 (*GPU*)
PRME	550	2	52	-
JNTM	1,125	107	82	9

从表4.11中，可以看到我们的内存占用大约是 PRME 的两倍。这主要是因为我们为每个用户设置了两个固定表示 (即好友和偏好表示)，而 PRME 只有一个偏好表示。JNTM 的时间复杂度是 $O(d^2|D| + d|V|)$，而 PRME 是 $O(d|D|)$，其中 d 是嵌

① https://www.tensorflow.org

入表示维度 (我们的实验中 $d = 50$)。所以 JNTM 的运行时间大约是 PRME 的 d 倍。尽管我们的模型比 PRME 具有更长的训练时间，但 JNTM 用于测试的时间成本几乎与 PRME 相当。JNTM 对于单次的位置推荐，平均只需要 30ms，可以很高效地在训练阶段之后提供在线推荐服务。此外，GPU 加速提供了 12 倍的训练速度，证明 JNTM 可以在硬件支持下进行快速的训练。

4.5　本章小结

在本章中，我们通过联合建模社交网络和移动轨迹，提出了一种新颖的神经网络模型。具体而言，我们的模型包括两个组成部分：社交网络的构建和移动轨迹的生成。我们首先采用网络嵌入方法构建社交网络。我们考虑了影响移动轨迹生成过程的四个因素，即用户访问偏好，好友的影响，短期序列上下文和长期序列上下文。为了刻画两个序列上下文，我们分别使用 RNN 和 GRU 模型来建模不同级别（即短期或长期）的移动轨迹中的序列相关性。最后，我们通过共享的用户网络表示来联合这两个部分。在下一个位置推荐和好友推荐任务中，我们的模型始终优于基线方法。在我们的方法中，网络结构和轨迹信息相互补充。因此，当网络结构或轨迹数据稀疏时，我们对基线方法的改进更为显著。

第 5 章　富信息网络典型应用问题——微观层面的信息传播预测

在推荐系统之外，传播预测也是一项典型的富信息网络应用。在过去十年中，信息的传播和预测引起了很多研究者的关注。大部分传播预测的工作旨在预测级联级别的宏观属性，如信息传播的最终规模。现有微观层面的传播预测模型主要聚焦在用户级别的建模。现有方法要么对一个用户如何被一个级联所影响作出了很强的假设，要么将问题限制在‘‘谁影响了谁（who infected whom）’’信息已经被明确标出的特定场景中。强假设过分简化了复杂的信息传播机制，使得这些模型无法更好地拟合现实世界的级联数据。此外，针对特定场景的方法无法推广到未观察到传播图的一般设置中。

为了解决前人工作的这些不足，在本章[①]中，我们提出了面向一般微观层面传播预测的神经传播模型（Neural Diffusion Model），简称为 NDM。NDM 基于相对宽松的假设，并采用深度学习技术，包括注意力机制和卷积神经网络进行级联建模。这两个优势使我们的模型能够超越之前方法的限制，更好地拟合传播数据，并推广到训练集以外的级联上。四个真实级联数据集上的传播预测任务的实验结果表明，我们的模型相比于基线方法，F1 值可以有 26% 的相对提升。

5.1　问题描述

信息传播在人们的日常生活中无处不在，例如谣言的传播，病毒的传染以及新思想和新技术的传播。其传播过程，也称为级联，已经在广泛的领域内得到了研究。虽然有些工作认为即使最终的级联规模也无法预测[101]，近期工作[102–104] 已经展示了预测级联的规模、增长和许多其他关键性质的能力。如今，级联的建模和预测在许多实际应用中发挥着重要作用，例如产品推荐[105–109]，流行病学[110,111]，社交网络[112–114] 以及新闻和观点的传播[115–117]。

大部分已有的传播预测工作专注于宏观属性的预测，例如分享某张照片的用户总数[102] 或者一个博客关注度的增长曲线[103]。然而，宏观传播预测是对于级联的粗略估计，且无法适用于图5.1中的微观层面预测问题。微观传播预测更注重用户级别而不是级联级别的建模和预测，比宏观预测的能力要强大得多，并且能够在实际应用中开发针对特定用户的策略。例如，在推广新产品期间，微观传播预

① 本章主要工作以 “Neural Diffusion Model for Microscopic Cascade Prediction’ 为题投稿至国际学术期刊 “IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE)’’。

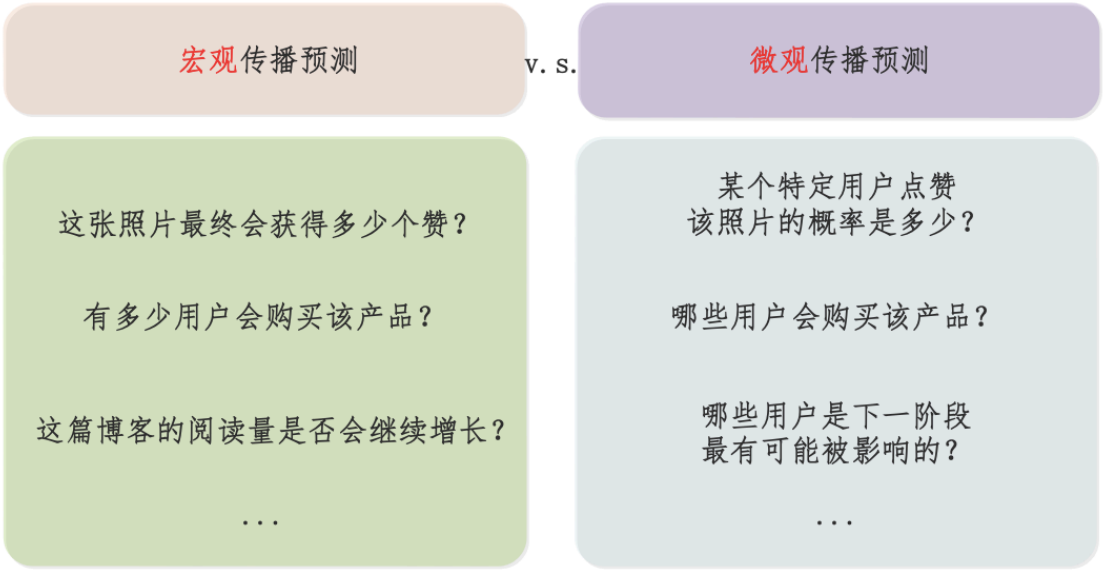


图 5.1　宏观传播预测 v.s. 微观传播预测

测可以帮助卖家向每个阶段最有可能购买该产品的用户推荐广告。本章工作专注于微观层面传播预测的研究。

虽然有用且强大，但级联的微观预测面临着巨大的挑战，因为现实世界的传播过程可能相当复杂[118] 而且观测通常是不完全的[111,119]：

复杂机制：因为一个具体用户如何被影响[①]的机制十分复杂，基于强假设和简单计算公式的传统传播模型可能并不是微观传播预测的最佳选择。现有的适用于微观层面预测的传播模型[120–123] 主要基于 Independent Cascade (IC) 模型[112]。IC 模型基于成对独立假设，为每个用户对 (u, v) 分配一个静态概率 $p_{u,v}$。$p_{u,v}$ 表示当用户 u 已经被激活时，用户 v 被 u 影响的可能性。其他传播模型[124,125] 甚至做出了更强的假设，即受影响用户只由源用户决定。虽然直观且易于理解，但这些级联模型都基于强假设和过度简化的概率估计公式，从而限制了模型的表达能力和适应复杂的真实数据的能力[126]。现实世界中信息传播的复杂机制鼓励我们为传播建模探索例如深度学习技术的更复杂的模型。

不完全观测：另一方面，级联数据通常都是不完全观测的，也就是说只能观测到用户被影响，而不知道是谁影响了这些用户。然而，据我们所知，现有的深度学习技术驱动的微观传播模型[127,128] 都是基于传播图已知的假设。这里传播图是指用户只能通过传播图中的边影响和被影响。举例来说，当研究 Twitter 网络中的转发行为时，‘‘谁影响了谁’’ 信息显式的存在于转发链中，且下一个可能被影响的用户被限制在邻居用户当中而不是所有用户。但是在大多数传播过程中，例如产品推广或病毒传染，传播图都是观测不到的[111,119,129]。因此，这些方法实际上考虑了

① 我们用 ‘‘影响’’ 或者 ‘‘激活’’ 来表示一个用户被级联所影响

一个更简单的问题，并不能推广到传播图未知的一般设置。

为了填补一般场景下微观层面传播预测的空白，并解决传统传播模型的局限性，我们采用了最新的深度学习技术，即注意力机制和卷积神经网络，并提出了一种基于更宽松的假设的神经传播模型用于级联建模。宽松的假设使得模型可以更灵活且少受约束，而深度学习模型擅长捕捉人工设计的特征难以表达的复杂内在关系。这两种优势使该模型能够超越基于强假设和过度简化的公式的传统方法的局限性，更适合复杂的级联数据。参照前人工作[123] 的实验设置，本章工作在四个真实的级联数据集上进行了传播预测任务的实验，以评估本章提出的模型和其他最先进的基线方法的性能。实验结果表明该模型相比于表现最好的基线方法，F1 值有 26% 的相对提升。

总结来说，本章工作有以下三个主要贡献：

(1) 据我们所知，本章工作首次尝试使用了深度学习技术来解决传播图未知的微观级别传播预测问题。

(2) 本章工作设计了一个神经传播模型，其模型假设比传统传播模型中的成对独立假设更宽松，可以更好地拟合现实世界的级联数据。

(3) 四个真实数据集上的传播预测任务的实验结果证明了本章提出的模型的有效性和鲁棒性。和最好的基线方法相比，该模型在 F1 值上取得了 26% 的相对提升。

5.2　相关工作

本章的相关工作主要分为宏观和微观层面传播预测方法。

5.2.1　宏观层面的传播预测

大部分已有的传播预测工作专注于宏观层面，例如一个级联的最终规模[104] 和增长趋势[103]。宏观传播预测方法可以进一步分为基于特征的方法、生成式的方法和深度学习方法。基于特征的方法将其看作分类[102,130] 或者回归[131,132] 问题，使用 SVM、逻辑回归和其他机器学习算法应用于人工设计的时序特征[133] 和结构特征[102]。生成式的方法将级联规模的增长看作受影响用户的到达过程，并利用例如 Hawkes 自激点过程[104,134] 等随机过程来建模。随着深度学习技术在各种应用中的成功，基于深度学习的方法，例如 DeepCas[126] 和 DeepHawkes[135]，提出使用循环神经网络 Recurrent Neural Network (RNN) 编码级联序列以取代人工设计的特征。与特征工程相比，基于深度学习的方法在不同场景上具有更好的泛化能力，在宏观预测任务上具有更好的性能。

5.2.2　微观层面的传播预测

本章的工作与微观传播预测更加相关。微观层面的传播预测侧重于用户级建模和预测。我们将相关工作分为三类：基于 IC 模型的方法、基于嵌入表示的方法和基于深度学习的方法。

IC 模型[112,115,136,137] 假设每对用户间的传播概率是独立的，是最普遍使用的传播模型之一。IC 模型的扩展通过引入预定义的时间衰减权重函数进一步考虑了时间延迟信息，例如 continuous time IC[138]，CONNIE[119]，NetInf[120] 和 Netrate[121]。Infopath[122] 提出基于信息传播数据推断动态传播概率并研究了信息路径的时间演化。MMRate[139] 通过研究不同方面下的用户行为和传播模式来学习多方面传播概率。以上所有方法都从级联序列中学习了用户间的传播概率。当模型完成训练时，我们可以通过蒙特卡罗模拟级联中用户序列的生成过程以进行微观传播预测。

基于嵌入表示的方法将每个用户编码为一个参数化的实值向量，并通过最大化目标函数来学习参数。Embedded IC[123] 遵循 IC 模型中的成对独立假设，并通过用户嵌入表示对两个用户之间的传播概率进行建模。其他基于嵌入表示的传播模型[124,125] 甚至做出了更强的假设，即受影响的用户仅由源用户和信息内容决定。如先前工作[126] 所示，这些基于强假设的模型过度简化了现实机制，通常在实际预测任务中表现不佳。

现有的基于深度学习的微观传播预测方法[127,128] 专注于社交网络中的转发和分享行为，其中 ‘‘谁影响了谁” 信息在转发链中已经被明确地标出。当传播图已知时，下一个受影响的用户候选仅限于邻居用户。然而，对于大多数传播过程，传播图通常是未知的[111,119]。例如，在某个病毒的传染过程中，我们无法观察到患者是被谁传染的病毒。现有的基于深度学习的方法考虑了相对简单的场景，并不能推广到未观察到传播图的一般设置下。据我们所知，我们的工作首次尝试了深度学习技术来解决传播图未知的一般微观层面传播预测问题。

5.3　模型框架

在这一节中，我们首先对实验中所用的数据集进行介绍和观察，然后提出两个基于该观察的模型假设，最后介绍基于这两个模型假设设计的神经传播模型（NDM）。

5.3.1　数据观察

在本小节中，我们将对真实数据集进行数据观察，并研究传播序列中被激活用户之间的内在关系。具体而言，我们将尝试研究被激活的连续用户之间是否有

更强的相关性，从而一起出现在更多的传播序列中。我们将首先介绍数据集。

5.3.1.1　数据集

我们收集了四个真实的级联数据集，涵盖了各种应用场景。级联是在一群用户之间传播的某种信息。每个级联包含了一个 (用户，时间戳) 对的序列，其中每对 (用户，时间戳) 表示了该用户在此时间点被激活的事件。

Lastfm[140] 是一个音乐流网站。数据集包含了近 1,000 名用户一年内听过的歌曲的完整历史。我们将每首歌当作一个用户间的传播对象并去掉了收听不多于 5 首歌的用户。

Irvine[141] 是加州大学 Irvine 分校学生的在线社区。学生可以在不同的论坛上参与和撰写帖子。我们将每个论坛当作一个传播对象并去掉了参加不多于 5 个论坛的用户。

Memetracker①收集了一百万条新闻报道和博客文章，并追踪其中最常见的引用和短语，即模因，来研究模因在人群中的传播。每个模因被看作一个传播对象而每个网站或博客的 URL 则被看作一个用户。遵循先前工作[123] 的设置，我们只保留了最活跃的一部分 URL 来避免噪音的影响。

Twitter 数据集[142] 收集了 2010 年 10 月 Twitter 上包含 URL 的微博信息，囊括了每个 URL 的完整微博历史。我们将每个不同的 URL 看作 Twitter 用户间的传播对象。我们过滤掉了不超过 5 条微博的不活跃用户。Twitter 数据集的规模和之前的基于神经网络的传播模型[123,128] 所使用的相当甚至更大。

上述所有数据集都没有明确的用户是被谁影响的信息。虽然我们在 Twitter 数据集中有用户间好友关系，但除非用户直接转发，否则我们仍然无法追踪某个用户发布特定 URL 的传播来源。

表 5.1　数据集统计

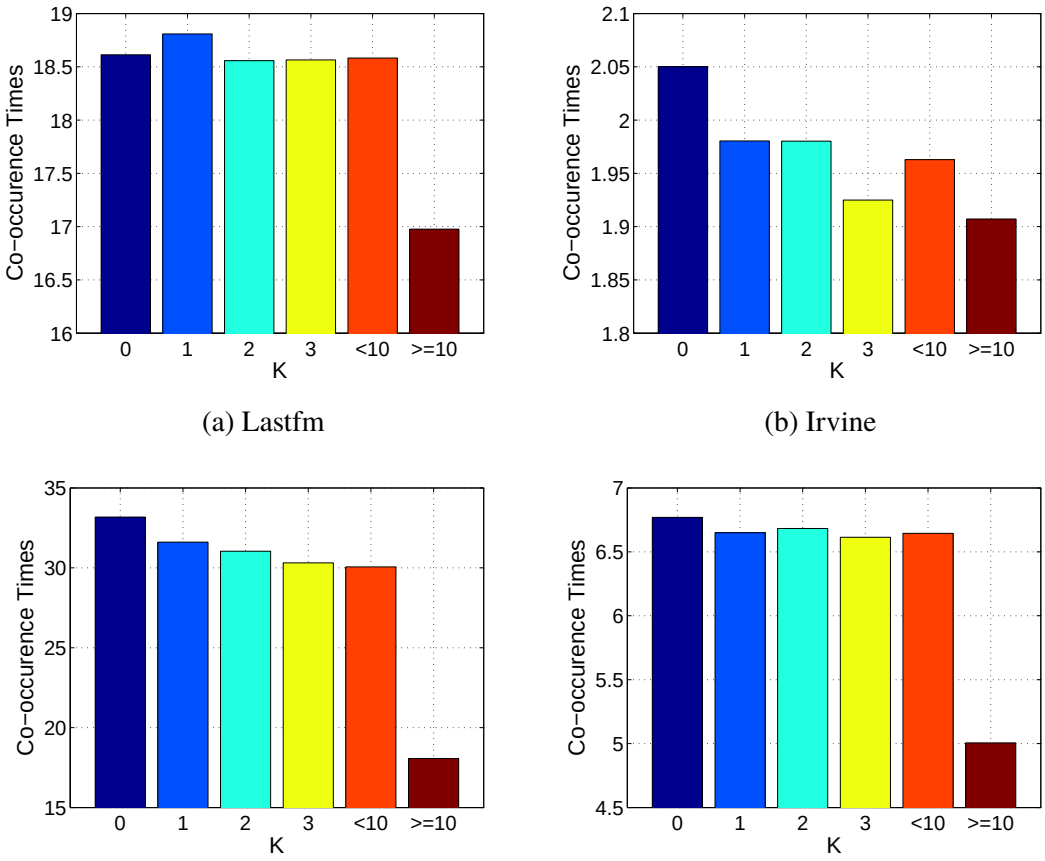
数据集	# 用户	# 链接	# 级联	平均长度
Lastfm	982	506,582	23,802	7.66
Irvine	540	62,605	471	13.63
Memetracker	498	158,194	8,304	8.43
Twitter	19,546	18,687,423	6,158	36.74

我们在表5.1中列出了数据集的统计信息。我们假设如果两个用户出现在同一级联序列中，那么这两个用户之间存在链接。在传统 IC 模型中，每条虚拟的 ‘‘链

① http://www.memetracker.org

接'' 都会被分配一个参数化的概率。因此，传统方法的空间复杂性相对较高，特别是对于大型数据集。我们还在最后一列中计算了每个数据集的平均级联长度。

5.3.1.2　统计分析



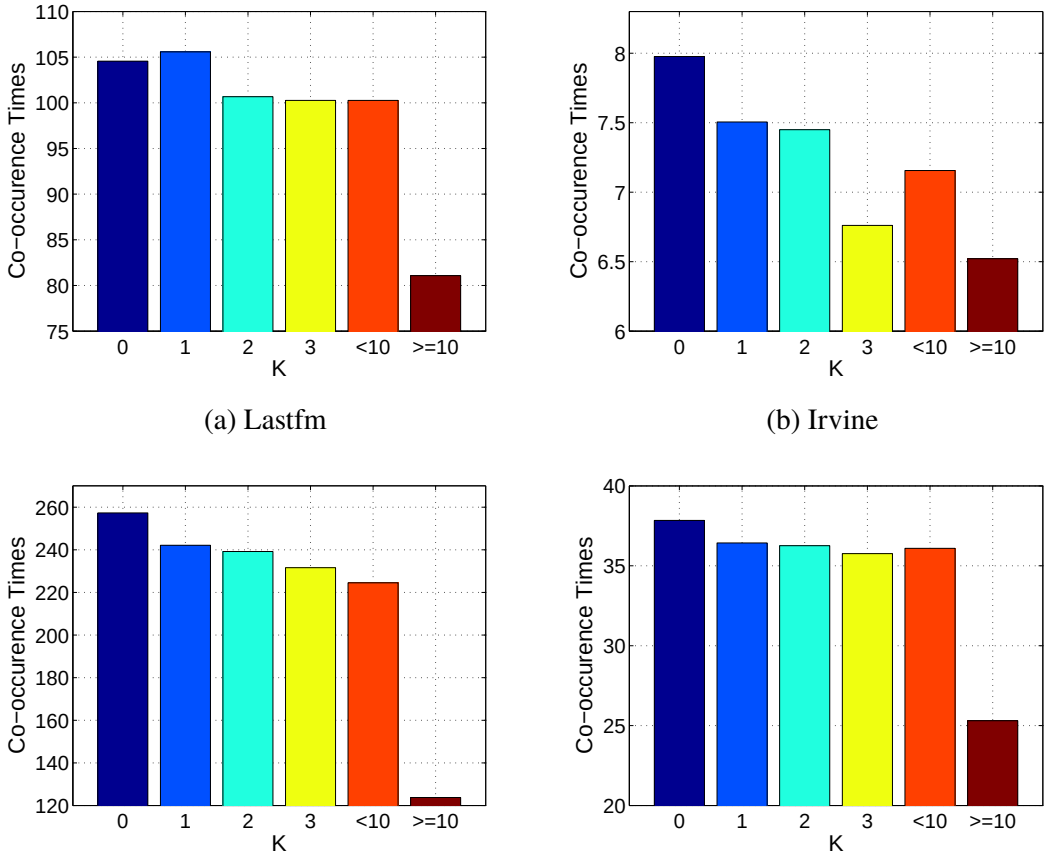
(a) Lastfm　(b) Irvine

(c) Memetracker　(d) Twitter

图 5.2　给定两个随机用户出现在同一级联中且中间间隔 K 个用户的条件下，这两个用户共现的级联数 (Co-occurrence Times) 的期望的统计结果。这里 < 10 和 ≥ 10 分别是对 $K < 10$ 和 $K \geq 10$ 的平均共现次数。

现在我们将尝试通过统计结果揭示用户之间的相关性。直觉上看，级联序列中的两个连续激活的用户更可能相关（例如前一个激活了后一个），从而一同出现在很多其他的传播序列中。

为了验证这个假设，我们考虑了以下统计实验：给定用户 u_i 和 u_j 出现在同一级联中且中间间隔 K 个用户的条件，那么用户 u_i 和 u_j 都参与的级联序列数的期望将是多少？这里 $K = 0$ 表示用户 u_i 和 u_j 是被连续激活的。如果我们的假设是正确的，那么这个期望将随 K 减小而增加。



(a) Lastfm　　(b) Irvine

(c) Memetracker　　(d) Twitter

图 5.3　给定两个随机用户出现在同一级联中且中间间隔 K 个用户的条件，且他们是满足前一条件的共现次数**前 5%** 的用户对，这两个用户共现的级联数 (Co-occurrence Times) 的期望的统计结果。

图5.2展示了全部四个数据集上的统计结果。这里我们列举了 $K = 0, 1, 2, 3$，$K < 10$ 和 $K \geq 10$ 的统计结果。统计表明 $K < 10$ 时共现次数的期望一致地大于 $K \geq 10$ 的。由于长尾效应，某些数据集的区别不是很大。因此，我们进一步在图5.3中展示了共现次数前 5% 的用户对的统计结果。我们可以在这个设置下看到显著的区别。

这些统计结果表明，级联序列中连续激活的用户更可能具有相关性。这里用户间的 ‘‘相关’’ 是指他们之间有直接的传播路径或者他们可能都是被同一用户激活。此外，我们发现不仅上个被激活的用户 ($K = 0$) 与当前用户相关：如图5.2和5.3所示，所有近期被激活的用户 ($K = 0, 1, 2, 3$) 都可能与当前用户相关（统计结果差异较小）。基于这些发现，我们将在下一小节中构建我们的模型。

5.3.2　模型构建

在本小节中，我们首先将形式化问题并介绍符号定义。然后，我们根据数据观察提出两个启发式假设作为基础，并使用深度学习技术设计神经传播模型（NDM）。最后，我们将介绍我们模型整体的优化目标函数和其他细节。

5.3.2.1　问题形式化

级联数据集记录了传播对象在其传播期间在何时传播给谁的信息。例如，传播对象可以是产品，级联记录了谁在何时购买了该产品。但是在大多数情况下，用户之间不存在明确的传播图[123,137]。因此，我们没有一个用户是如何被其他用户影响的具体信息。

形式化地，给定用户集合 $\mathcal{U}$ 和观察到的级联序列集合 C，每个级联 $c_i \in C$ 包含一个按被影响时间排序的用户列表 $\{u_0^i, u_1^i \dots u_{|c_i|-1}^i\}$，其中 $|c_i|$ 是序列 c_i 的长度，$u_j^i \in \mathcal{U}$ 是序列 c_i 中的第 j 个用户。在本章工作中，我们像之前的工作一样[123,128,129]，只考虑了用户被影响的顺序，并忽略了对应的确切时间信息。

在本章中，我们的目标是学习给定不完全观测的级联序列 $\{u_0, u_1 \dots u_j\}$ 时，能够预测下一个被影响用户 u_{j+1} 的传播预测模型。学习到的模型能够基于前几个观察到的受影响用户来预测整个受影响的用户序列并用于图5.1中的微观传播预测任务。在我们的模型中，我们在用户集合 $\mathcal{U}$ 中加入了名为 <STOP> 的虚拟用户。在训练阶段，我们在每个级联序列的末尾添加 <STOP> 来表示在这个级联中不会有更多的用户被影响。

更进一步，我们通过参数化的实值向量来表示每个用户，以将用户映射到向量空间中。实值向量也称为嵌入。我们将用户 u 的嵌入表示记为 $emb(u) \in \mathbb{R}^d$，其中 d 是嵌入维度。在我们的模型中，两个用户的嵌入表示之间的内积越大表明用户之间的相关性越强。如图5.4所示，嵌入表示层将用户映射为对应的向量表示，是我们模型的最底层。

5.3.2.2　模型假设

在传统的 Independent Cascade (IC) 模型[112] 设置中，所有已经受影响的用户无论他们受影响的顺序如何，都可以独立且平等地激活新用户。很多 IC 模型的扩展进一步考虑了时间衰减信息，例如 continuous time IC (CTIC)[138] 和 Netrate[121]。但是，这些模型都没有去尝试找出在当前时刻更有可能激活其他用户的真正的活跃用户。为解决这个问题，我们提出了以下假设。

假设 1：给定最近被影响的用户 u，和用户 u 强相关的用户包括 u 本身更可能是活跃的。

这一假设是直观的。作为新激活的用户，u 应该是活跃的并有可能影响其他用户。和用户 u 强相关的用户很可能是 u 被激活的原因，所以在此时和其他用户相比更可能是活跃的。我们进一步提出 ‘‘活跃用户表示’’ 概念来建模所有这样的活跃用户。

定义 1：对于每个近期被影响的用户 u，我们目标是学习一个活跃用户表示 $act(u) \in \mathbb{R}^d$，即所有和用户 u 相关的活跃用户的嵌入表示，并将其用于预测下一个受影响的用户。

活跃用户表示 $act(u_j)$ 刻画了和用户 u_j 被激活相关的潜在活跃用户。从数据观察中，我们可以看到所有最近被影响的用户都可能与下一个受影响的用户相关。因此，所有最近激活的用户的活跃用户表示都应该有助于预测下一个受影响的用户，从而引出以下假设。

假设 2：所有最近被影响的用户都应该贡献于下一个受影响用户的预测，并根据激活的顺序进行不同的处理。

和相关工作中介绍的基于 IC 和嵌入表示的模型中的强假设相比，我们的启发式假设使我们的模型更加灵活，可以更好地拟合级联数据。现在我们将介绍如何基于这两个假设来构建我们的模型，即抽取活跃用户并结合其嵌入表示以进行预测。

5.3.2.3　用注意力机制抽取活跃用户

为了计算活跃用户表示，我们提出使用注意力机制[143,144] 来抽取最可能的活跃用户。注意力机制会给予活跃用户比其他用户更多的权重。如图5.4所示，用户 u_j 的活跃用户表示由先前被影响用户的加权和计算得到：

$$act(u_j) = \sum_{k=0}^{j} w_{jk} emb(u_k), \tag{5-1}$$

其中用户 u_k 的权重为

$$w_{jk} = \frac{\exp(emb(u_j) emb(u_k)^T)}{\sum_{m=0}^{j} \exp(emb(u_j) emb(u_m)^T)}. \tag{5-2}$$

注意对于所有 k 有 $w_{jk} \in (0, 1)$ 且 $\sum_{m=0}^{j} w_{jm} = 1$。$w_{jk}$ 是 u_j 和 u_k 的嵌入表示的归一化的内积，表示了 u_j 和 u_k 之间相关性的强度。

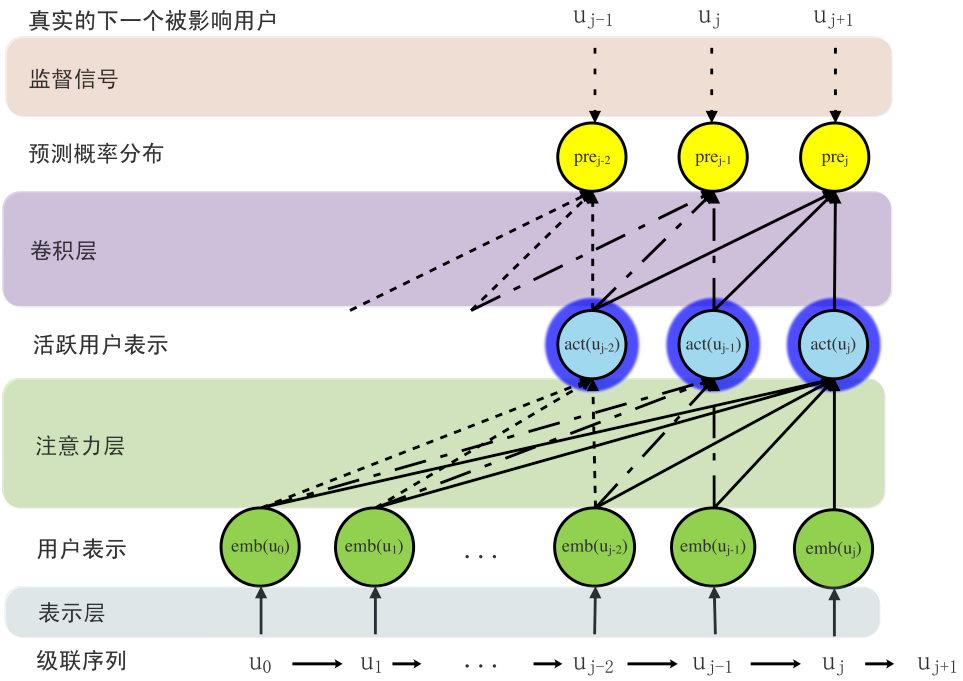


图 5.4　神经传播模型 (NDM) 示意图。NDM 基于最近被激活用户的活跃用户表示（蓝色节点）顺序地预测下一个被影响的用户，并利用所有先前被影响用户的用户表示（绿色节点）上的注意力层计算活跃用户表示。

从公式（5-1）中定义的活跃用户表示 $act(u_j)$，我们可以看到和 $emb(u_j)$ 内积越大的 $emb(u_k)$ 会被分配越大的权重 w_{jk}。这个公式遵循了我们的假设：和用户 u 强相关的用户包括 u 本身应该得到更大的关注。

为了充分利用神经模型的优势，我们进一步使用了多头注意力机制[144] 来提高表达能力。多头注意力用不同的线性变换将用户表示映射到多个子空间中。然后多头注意力独立地在每个子空间内执行注意力机制。最后，多头注意力将所有子空间中的注意力表示拼接起来，并再次对结果进行线性变换。

形式化地，在有 h 个头的多头注意力机制中，第 i 个头的注意力表示为

$$head_i = \sum_{k=0}^{j} w_{jk}^i emb(u_k) W_i^V, \tag{5-3}$$

其中

$$w_{jk}^i = \frac{\exp(emb(u_j)W_i^Q(emb(u_k)W_i^K)^T)}{\sum_{m=0}^{j}\exp(emb(u_j)W_i^Q(emb(u_m)W_i^K)^T)}, \tag{5-4}$$

$W_i^V, W_i^Q, W_i^K \in \mathbb{R}^{d\times d}$ 是各个头的线性变换矩阵。具体地，W_i^Q 和 W_i^K 可以看作将用户表示分别映射到*接收者*空间和*发送者*空间，以便进行非对称建模。

我们有活跃用户表示 $act(u_j)$

$$act(u_j) = [head_1, head_2 \dots head_h]W^O, \tag{5-5}$$

其中 [] 表示拼接操作，$W^O \in \mathbb{R}^{hd\times d}$ 将拼接后的结果映射到 d 维向量空间。

多头注意力机制可以让模型从不同角度（子空间）独立地处理信息，从而比传统的注意力机制更加强大。

5.3.2.4　用卷积神经网络联合活跃用户表示进行预测

和直接设置时间衰减权重的先前工作[121,145] 不同，我们提出使用参数化神经网络来处理不同位置的活跃用户表示。和预定义的指数衰减权重相比[121]，参数化的神经网络可以自动学习并拟合现实世界数据集并捕捉每个位置的活跃用户表示与下一个受影响用户预测之间的内在关系。在本章中，我们考虑使用卷积神经网络（CNN）来实现这一目的。

CNN 已经在计算机视觉[146]，推荐系统[147] 和自然语言处理[148] 等领域中得到了广泛应用。CNN 是*平移不变*的神经网络，使得我们可以分配特定于位置的线性变换。

图5.4给出了我们的卷积层窗口大小 $win = 3$ 时的示意图。卷积层首先用位置特定的线性变换矩阵 $W_n^C \in \mathbb{R}^{d\times|\mathcal{U}|}, n = 0, 1 \dots win-1$，将每个活跃用户表示 $act(u_{j-n})$ 转换为 $|\mathcal{U}|$ 维的向量。然后卷积层对转换后的向量求和并用 *softmax* 函数对求和进行归一化。

形式化地，给定不完全观测的级联序列 $(u_0, u_1 \dots u_j)$，预测概率分布 $pre_j \in \mathbb{R}^{|\mathcal{U}|}$ 为

$$pre_j = \text{softmax}(\sum_{n=0}^{win-1} act(u_{j-n})W_n^C), \tag{5-6}$$

其中 $\text{softmax}(x)[i] = \frac{\exp(x[i])}{\sum_p \exp(x[p])}$，$x[i]$ 表示向量 x 的第 i 维。pre_j 的每一维代表对应用户在下一步被影响的概率。

因为初始用户 u_0 在整个传播过程中扮演着重要的角色，我们进一步考虑了 u_0：

$$pre_j = \text{softmax}(\sum_{n=0}^{win-1} act(u_{j-n})W_n^C + act(u_0)W_{init}^C \cdot F_{init}), \tag{5-7}$$

其中 $W_{init}^{C} \in \mathbb{R}^{d\times|\mathcal{U}|}$ 是初始用户 u_0 的映射矩阵，$F_{init} \in \{0,1\}$ 是控制是否将初始用户加入预测的超参数。

5.3.2.5　整体架构，模型细节和学习算法

我们自然地最大化所有观察到的级联序列的对数似然作为整体优化目标：

$$\mathcal{L}(\Theta) = \sum_{c_i \in C} \sum_{j=0}^{|c_i|-2} \log pre_j^i[u_{j+1}^i], \tag{5-8}$$

其中 $pre_j^i[u_{j+1}^i]$ 是级联 c_i 的第 j 个位置的真实的下个被影响用户 u_{j+1}^i 的概率，Θ 是需要学习的所有参数的集合，包括每个用户 $u \in \mathcal{U}$ 的嵌入表示 $emb(u) \in \mathbb{R}^d$，多头注意力机制的线性变换矩阵 $W_n^V, W_n^Q, W_n^K \in \mathbb{R}^{d\times d}, n = 1,2\ldots h, W^O \in \mathbb{R}^{hd\times d}$ 以及卷积层的矩阵 $W_{init}^C, W_n^C \in \mathbb{R}^{d\times|\mathcal{U}|}, n = 0,1\ldots \text{win}-1$。

实现细节：我们用 PyTorch[①]实现模型和 Adam optimizer[149] 计算梯度学习参数。我们进一步在计算活跃用户表示时使用了层归一化[150] 和残差连接[151] 以避免在深度神经网络中可能发生的梯度爆炸或消失问题。具体地，活跃用户表示 $act(u)$ 被替换为 $LayerNorm(emb(u)+act(u))$，其中 $LayerNorm(\cdot)$ 倾向于使输出拥有零均值和单位方差。我们也在注意力机制中使用了 dropout[152] 来避免过拟合，dropout 率被设为 0.1。因为同一个用户不会被影响两次，我们在公式（5-7）中去掉了已经被影响的用户。我们的实现代码公开在github[②]，超参数将在下节中介绍。

复杂度：我们模型的空间复杂度是 $O(d|\mathcal{U}|)$，其中 d 是嵌入维度，远小于用户集的大小。注意训练传统 IC 模型的空间复杂度会有 $O(|\mathcal{U}|^2)$ 因为我们需要对每对有潜在联系的用户对分配一个影响概率。因此，我们的神经传播模型的空间复杂度低于传统的 IC 方法。

每个活跃用户表示的计算需要 $O(|c_i|d^2)$ 时间，其中 c_i 是对应级联的长度。公式（5-7）中的下一受影响用户概率预测需要 $O(d|\mathcal{U}|)$ 时间。所以训练一个级联的时间复杂度是 $O(\sum_{c_i\in C}(|c_i|^2d^2+|c_i|d|\mathcal{U}|))$，和已有的神经网络模型（例如 embedded IC 模型[123]）相当。但正如我们将在实验中展示的，我们的模型比 embedded IC 模型收敛得更快，并且能够处理大规模数据集。

① http://pytorch.org

② https://github.com/albertyang33/NeuralDiffusionModel

5.4　实验结果

遵循先前工作[123] 的设置，我们用微观层面的传播预测实验来评估我们的模型和其他基线方法的表现。我们首先介绍基线方法，评估指标和超参数设置。然后我们将展示实验结果并进一步分析评估。

5.4.1　基线方法

我们考虑了许多最先进的基线方法来证明我们的算法的有效性。大多数基线方法会从级联序列中学习转移概率矩阵 $M \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{U}|}$，其中 M_{ij} 表示当用户 u_i 被影响时，u_j 被 u_i 影响的概率。

Netrate[121] 考虑了传播概率的时变动态性，并定义了指数、幂律和瑞利三种传播概率模型，它们促使传播概率随着时间间隔的增加而减小。在我们的实验中，我们只报告指数模型的结果，因为其他两个模型结果类似。

Infopath[122] 也基于传播数据推断动态的传播概率。Infopath 采用随机梯度来估算时间动态并研究了信息路径的时间演变。

Embedded IC[123] 利用表示学习技术，通过用户嵌入表示的函数而不是静态值来建模两个用户之间的传播概率。Embedded IC 模型采用随机梯度下降法进行训练。

LSTM 是一个面向序列建模的广泛使用的神经网络框架[153]，并在最近被用于级联建模。先前工作在一些更简单的任务场景中利用了 LSTM，例如规模预测[126] 和概率图已知的传播预测[127,128]。因为这些算法无法直接和我们的模型进行比较，我们通过在 LSTM 的隐状态上添加 softmax 分类器来让 LSTM 网络适用于下一受影响用户预测。

5.4.2　神经网络模型的超参数设置

虽然基于神经网络的方法的参数空间远小于传统的 IC 模型，我们必须设置几个超参数来训练神经模型。为了调整超参数，我们随机选择 10% 的训练级联作为验证集。

对 Embedded IC 模型，和原始论文[123] 一样，用户表示维度从 $\{25, 50, 100\}$ 中选取。对 LSTM 模型，用户表示和隐状态维度设为 $\{16, 32, 64, 128\}$ 中的最佳值。对于我们的模型 NDM，多头注意力中的头数 $h = 8$，卷积层窗口大小 $win = 3$，用户表示维度 $d = 64$。对于所有数据集，我们使用同样的 (h, win, d)。公式（5-7）中的 F_{init} 决定了初始用户是否用于预测：对于 Twitter 数据集，$F_{init} = 1$；对于其他三个数据集，$F_{init} = 0$。我们将在参数敏感性小节中展示我们模型的鲁棒性。

神经网络模型，即 Embedded IC，LSTM 和 NDM，都是基于矩阵乘法运算，可以通过 GPU 进行加速。因此，我们在 GPU 设备上 (GeForce GTX TITAN X) 而不是 CPU 设备上 (Intel Xeon E5-2620 @ 2.0GHz) 训练这三个模型。

5.4.3　微观级别的传播预测

为了比较级联建模的能力，我们在微观传播预测任务上评估我们的模型和所有基线方法。我们采用 Embedded IC[123] 中的实验设置，随机选取了 90% 的级联序列作为训练集，其余的作为测试。对测试集中的每个级联序列 $c = (u_0, u_1, u_2 \dots)$，只有初始用户 u_0 已知，所有后续被影响用户 $G^c = \{u_1, u_2 \dots u_{|G^c|}\}$ 需要被预测。

所有基线方法和我们的模型需要预测一个用户集合，并将结果与 实际受影响的用户集合 G 进行比较。对于基于 IC 模型的基线方法，即 Netrate，Infopath 和 Embedded IC，我们将根据学习得到的用户间传播概率及其相应的生成过程来模拟传播过程。对于 LSTM 和我们的模型，我们将根据 softmax 分类器的概率分布顺序地采样用户。

注意实际被影响的用户集合也可能是不完全观测的，因为数据集都是在一个相对较短的时间窗口内爬取的。因此，对于每个包含 $|G^c|$ 个受影响用户的测试级联 c，所有的算法在一次模拟中只需要预测前 $|G^c|$ 个受影响的用户。同时注意模拟过程可能会在激活 $|G^c|$ 个用户前终止。

对于所有算法，我们对每个测试级联序列 c 进行了 1000 次的蒙特卡洛模拟，并计算了每个用户 $u \in \mathcal{U}$ 被传播到的概率 P_u^c。我们用两个经典的评价指标 Macro-F1 和 Micro-F1 作为评价标准。

Macro-F1. Macro-averaged F1 首先对每个测试集级联序列 C_{test} 中的级联 c 计算准确率 pre_c，召回率 rec_c 和 F1 值 f_c。然后对所有的级联取平均：

$$pre_c = \frac{\sum_{u \in G^c} P_u^c}{\sum_{u \in \mathcal{U}} P_u^c}, rec_c = \frac{\sum_{u \in G^c} P_u^c}{|G^c|}, f_c = \frac{2pre_c \cdot rec_c}{pre_c + rec_c},$$

$$Macro - F1 = \frac{\sum_{c \in |C_{test}|} f_c}{|C_{test}|}.$$

Micro-F1. Micro-averaged F1 通过在所有的预测上取平均，全局地计算准确率 pre，召回率 rec。Micro-F1 通过为更长的级联分配更大的权重，可以作为 Macro-F1

表 5.2　微观级别传播预测的实验结果

评价指标	数据集	方法					相对提升
		Netrate	Infopath	Embedded IC	LSTM	NDM	
Macro-F1	Lastfm	0.017	0.030	0.020	0.026	**0.056**	+87%
	Memetracker	0.068	0.110	0.060	0.102	**0.139**	+26%
	Irvine	0.032	0.052	0.054	0.041	**0.076**	+41%
	Twitter	-	0.044	-	0.103	**0.139**	+35%
Micro-F1	Lastfm	0.007	0.046	0.085	0.072	**0.095**	+12%
	Memetracker	0.050	0.142	0.115	0.137	**0.171**	+20%
	Irvine	0.029	0.073	0.102	0.080	**0.108**	+6%
	Twitter	-	0.010	-	0.052	**0.087**	+67%

表 5.3　只预测每个级联前 5 个用户的早期微观级别传播预测的实验结果

评价指标	数据集	方法					相对提升
		Netrate	Infopath	Embedded IC	LSTM	NDM	
Macro-F1	Lastfm	0.018	0.028	0.010	0.018	**0.048**	+71%
	Memetracker	0.071	0.094	0.042	0.091	**0.122**	+30%
	Irvine	0.031	0.030	0.027	0.018	**0.064**	+106%
	Twitter	-	0.040	-	0.097	**0.123**	+27%
Micro-F1	Lastfm	0.016	0.035	0.013	0.019	**0.045**	+29%
	Memetracker	0.076	0.106	0.040	0.094	**0.126**	+19%
	Irvine	0.028	0.030	0.029	0.020	**0.065**	+117%
	Twitter	-	0.050	-	0.093	**0.118**	+27%

的补充：

$$pre = \frac{\sum_{c\in|C_{test}|}\sum_{u\in G^c} P_u^c}{\sum_{c\in|C_{test}|}\sum_{u\in \mathcal{U}} P_u^c}, rec = \frac{\sum_{c\in|C_{test}|}\sum_{u\in G^c} P_u^c}{\sum_{c\in|C_{test}|}|G^c|},$$

$$Micro-F1 = \frac{2pre\cdot rec}{pre+rec}.$$

为了进一步评估级联早期预测的性能，我们通过仅预测每个测试级联中的前五个受影响用户进行了补充实验。我们在表5.2和5.3中展示了实验结果。这里 “-’’ 表示算法没有在 72 小时内收敛。最后一列表示 NDM 相对于最好的基线方法的相对提升。我们有如下观察：

(1) NDM 显著且一致地超过所有基线方法。如表5.2所示，就 Macro-F1 值而

言，NDM 相比于最好的基线方法的提升至少有 26%。Micro-F1 值的提升进一步证明了我们提出的模型的有效性和鲁棒性。结果还表明，精心设计的神经网络模型能够超越传统的级联建模方法。

(2) NDM 在早期传播预测上有更加显著的提升。如表5.3所示，NDM 在 Macro 和 Micro F1 值上都远远超过了所有基线方法。在实际应用中，准确地预测第一批受影响用户非常重要，因为错误的预测会导致后续阶段的错误传播。在传播早期对受影响用户的精确预测可以使我们通过用户更好地控制信息的传播。例如，我们可以通过提前警告最脆弱的用户来防止谣言的传播或者通过向最有潜力的客户提供广告来促进产品的推广。该实验表明 NDM 具有用于实际应用的能力。

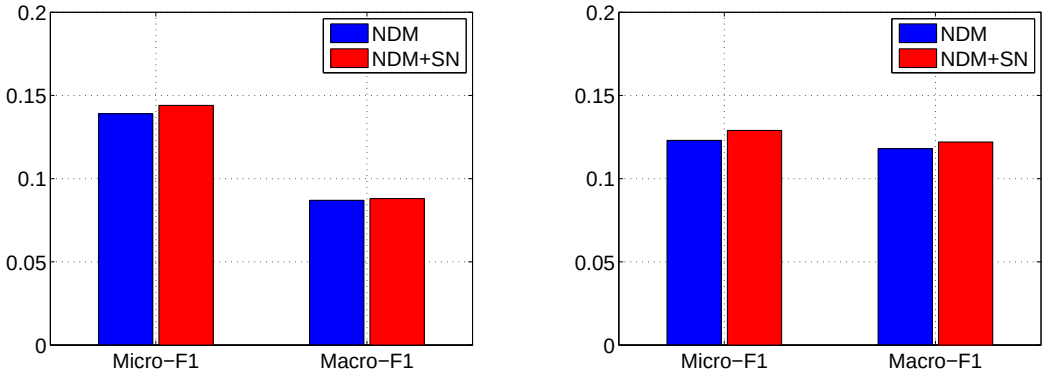
(3) NDM 可以应用于大规模数据集。Embedded IC 在拥有约两万用户和 1900 万潜在链接的 Twitter 数据集上没能在 72 小时内收敛。相反，NDM 在同样的 GPU 设备上可以在 6 小时内收敛，是 Embedded IC 速度的至少十倍。这一观察证明了 NDM 的高效。

5.4.4　额外的社交网络信息

有时用户间的社交网络是可以观测到的，如我们实验中所用的 Twitter 数据集。在 Twitter 数据集中，虽然信息传播并不一定通过社交网络中的好友关系，但是我们仍然希望传播预测过程可以从观察到的社交网络结构中受益。因此，我们对 NDM 模型进行了简单的修改以利用社交网络信息。现在我们将详细介绍。

首先，我们用 DeepWalk[154]，一个广泛使用的网络表示学习算法，将社交网络的拓扑结构映射为实值用户特征。DeepWalk 学习的网络表示维度设置为 32，即我们的模型维度 $d = 64$ 的一半。其次，我们用学习得到的网络表示来初始化我们模型中用户表示的前 32 维，并在后续的训练过程中保持不变。换句话说，64 维的用户表示由 DeepWalk 从网络结构中学习的 32 维的固定向量和 32 维随机初始化的向量构成。我们将结合了社交网络（Social Network）信息的模型命名为 NDM+SN。这是一个简单但实用的改动，我们将在未来的工作中探索更复杂的模型。图5.5展示了 NDM 和 NDM+SN 的对比。

实验结果表明 NDM+SN 借助于将社交网络结构作为先验知识，能够略微提高传播预测任务的性能。Micro-F1 的相对提升大约在 4%。实验结果表明，我们的神经传播模型非常灵活，可以方便地扩展以利用外部特征。



(a) 预测所有用户的微观传播预测　　(b) 只预测前 5 个用户的早期微观传播预测

图 5.5　NDM 和 NDM+SN 的预测效果对比

5.4.5　宏观规模预测

虽然我们提出的模型旨在进行微观层面的传播预测，但我们将在本小节中探索我们模型对于宏观级联规模预测的能力。在级联规模预测任务上，所有的模型需要在给定级联前 5 个受影响用户的条件下，预测级联的最终规模。我们使用了和微观预测中一样的数据集和数据划分。

注意在训练阶段，我们在每个级联序列的末尾添加了名为 <STOP> 的虚拟用户。当模型预测下一个受影响用户为 <STOP> 时，这表示该级联中不会有更多的被影响用户。和5.4.3小节的实验设置类似，我们对每个测试集中的级联序列进行蒙特卡罗模拟，以预测级联的最终规模。我们用现有级联规模预测方法[126,135] 中使用的 Mean Square Log-transformed Error (MSLE) 作为评价指标。具体地，MSLE= $\frac{1}{|C|}\sum_{i=1}^{|C|}(\log(|c_i|) - \log(pred_i))^2$，其中 $pred_i$ 是级联 c_i 的预测规模。

我们考虑了最先进的级联规模预测算法 DeepCas[126] 作为基线方法。实验结果见图5.6。

实验结果表明，即使 NDM 没有对宏观级别的预测进行直接的优化，但 NDM 在级联规模预测任务上拥有和 DeepCas[126] 相当甚至更好的性能。与将级联规模预测看作回归问题的 DeepCas 相比，NDM 实际上在训练数据中使用了更多信息，即具体的受影响用户以及他们的激活顺序。总之，我们提出的微观传播模型也可用于宏观的级联规模预测。NDM 利用更多信息实现了良好的性能。这些观察结果指出了宏观级别传播预测的未来方向，并启发我们设计用于微观和宏观预测的统一级联模型。

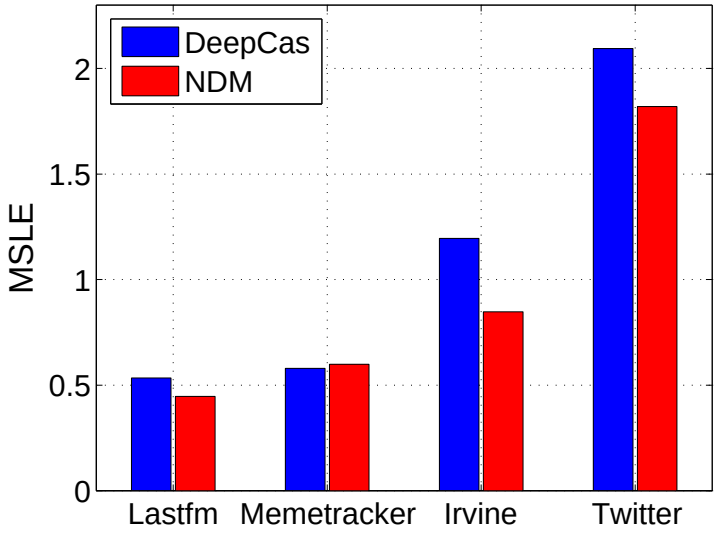


图 5.6　级联规模预测任务实验结果。MSLE 越小越好

5.4.6　参数敏感性

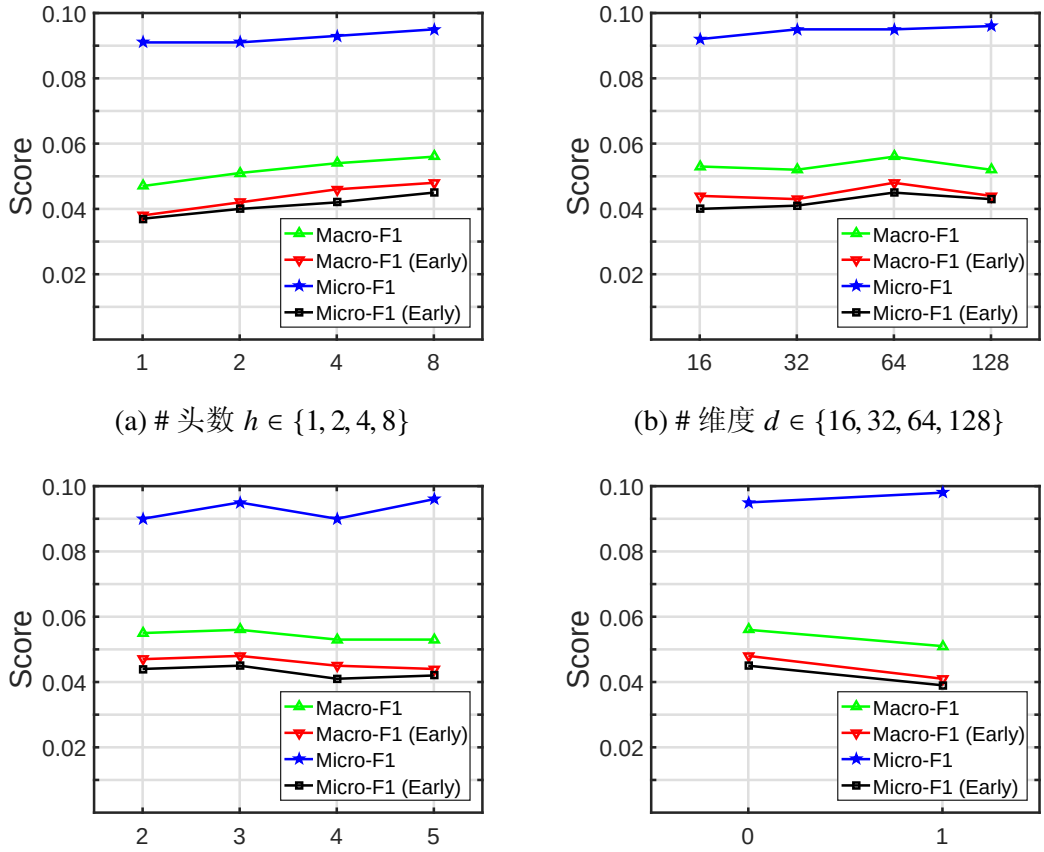
在本小节中，我们以 Lastfm 数据集为例来说明超参数设置如何影响我们模型的性能。我们使用最佳的超参数设置作为基础，即头数 $h = 8$，卷积网络窗口大小 $win = 3$，用户表示维度 $d = 64$ 和是否使用初始用户进行预测 $F_{init} = 0$。然后我们改变每个超参数，同时保持其他参数不变。图5.7展示了不同超参数设置下传播预测的性能。

我们可以看到，当我们在一个合理的范围内改变超参数时，NDM 的性能是相对稳定的。当我们将嵌入表示的维度 d 加倍到 128 时，NDM 不会遇到严重的过度拟合问题。这个实验证明了我们模型的鲁棒性。

5.4.7　可解释性

不可否认，可解释性通常是神经网络模型的一个弱点。与特征工程方法相比，基于神经的模型将用户编码为实值向量表示，并且用户表示的每个维度都没有明确的含义。在我们的模型中，每个用户表示被 8 头的注意力机制映射到 16 个子空间中。直觉上看，每个子空间中的用户表示都代表了一种用户角色。但我们很难将 16 种表示和可理解的人工设计特征联系起来。我们将在未来的工作中基于联合模型考虑用户表示和可解释特征之间的对齐。

幸运的是，我们在卷积层中能够有所发现。注意 $W_n^C \in \mathbb{R}^{d \times |\mathcal{U}|}, n = 0, 1, 2$ 是卷积层中位置特定的线性变换矩阵，W_{init}^C 是初始用户的映射矩阵。所有四个矩阵在训练之前随机初始化。在训练后的模型中，如果某个矩阵的数值尺度远大于其他



(a) # 头数 $h \in \{1, 2, 4, 8\}$　(b) # 维度 $d \in \{16, 32, 64, 128\}$

(c) # 窗口大小 $win \in \{2, 3, 4, 5\}$　(d) $F_{init} \in \{0, 1\}$

图 5.7　Lastfm 数据集上超参数的参数敏感性实验。Macro-F1 (Early) 和 Micro-F1 (Early) 对应早期预测的实验结果

矩阵，则预测向量更可能由相应位置的信息所支配。举例来说，如果 W_0^C 的数值尺度比其他矩阵更大，那么我们可以推断，最近的一个受影响用户对下个受影响用户预测贡献最大。

在这个实验中，我们对所有数据集设置 $F_{init} = 1$，并计算了所有映射矩阵的 Frobenius 范数。如表5.4所示，我们有如下观察：

(1) 对于所有数据集，W_0^C, W_1^C 和 W_2^C 大致相当且 W_0^C 总是比其他两个稍大一点。这一观察表明，最近三个受影响用户的活跃用户表示 $act(u_j), act(u_{j-1}), act(u_{j-2})$ 都对下个受影响用户 u_{j+1} 的预测有贡献。此外，最近的受影响用户 u_j 是三个中最重要的。这一发现和我们的直觉相符，且验证了方法一节中提出的假设 *2*。

(2) W_{init}^C 在 Twitter 数据集上最大。这表明初始用户在 Twitter 上的传播过程中非常重要。这可能是因为 Twitter 数据集包含了 URL 传播的完整历史记录，而初始用户实际上是第一个发布该 URL 的人。而在其他三个数据集中，初始用户只是爬取数据的时间窗口内的第一个用户。在传播预测任务中，我们只对 Twitter 数据集

表 5.4　通过 Frobenius 范数 $||\cdot||_F^2$ 测量的卷积层中映射矩阵的数值尺度

数据集	W_{init}^C	W_0^C	W_1^C	W_2^C
Lastfm	32.3	60.0	49.2	49.1
Memetracker	13.3	16.6	13.3	13.0
Irvine	13.9	13.9	13.7	13.7
Twitter	130.3	93.6	91.5	91.5

设置 $F_{init} = 1$，因为我们发现在其他三个数据集上，如果我们设置 $F_{init} = 1$，表现基本相当甚至更差。

5.5　本章小结

在本章中，我们提出了一种用于微观层面级联建模的神经传播模型（NDM）。为了超越基于强假设和过度简化公式的传统级联模型的局限性，我们基于两个启发式假设构建模型，并采用深度学习技术，包括卷积神经网络和注意力机制来实现假设。传播预测任务的实验结果证明了我们提出的模型的有效性和鲁棒性。此外，NDM 在早期传播预测上显著地超过了基线方法，展示了 NDM 在实际应用中的适用性和可行性。

第 6 章　富信息网络典型应用问题——多层面的信息传播预测

信息传播预测是研究信息如何在用户之间传播的重要任务。随着深度学习技术的成功，循环神经网络（RNN）已经展示出将信息传播建模为序列数据的强大能力。然而，之前的工作专注于预测下一个受影响的用户的微观传播预测或者估算传播过程中受影响用户的总数的宏观传播预测。据我们所知，现有工作没有尝试研究过能够同时进行微观层面和宏观层面预测的统一模型。在本章[①]中，我们提出了一种基于强化学习（RL）的多层面信息传播预测模型。强化学习通过解决梯度后向传播中的不可导问题，将宏观传播规模信息纳入基于 RNN 的微观传播模型。本章工作还采用了有效的结构上下文提取策略来利用数据中的社交网络结构信息。实验结果表明，本章工作提出的模型在三个真实世界数据集的微观和宏观传播预测上都优于最先进的基线模型。

6.1　问题描述

信息传播预测，也称为级联预测，已经在广泛的应用场景中得到了研究，例如产品推广[107–109]，流行病学[111]，社交网络[113,114] 以及新闻和观点的传播[116,117]。近期传播预测的工作[126,135,155,156] 利用深度学习技术的成功，将信息传播建模为基于循环神经网络（RNN）的序列数据，并取得了很好的成果。当信息通过社交网络服务进行传播时，现有工作[128,156] 也考虑了可用的社交网络信息帮助预测。

但是如图6.1所示，之前的工作要么专注于预测下一个受影响的用户的微观传播预测，要么致力于估算传播过程中受影响用户的总数的宏观传播预测。据我们所知，现有工作没有尝试研究过能够同时进行微观层面和宏观层面预测的统一模型。统一的模型可以利用训练数据中的更多信息，尤其是对于宏观传播预测而言。举例来说，先前工作[126,135] 将级联规模预测视为回归问题，并忽略了具体受影响用户及其受影响顺序的信息。

另外，现有工作[128,156] 对于社交网络信息的处理可能并不是最好的选择。TopoLSTM[128] 只考虑了社交网络中直接相连的用户对，而 SNIDSA[156] 计算了所有用户两两之间的相似度，承受着平方级的时间和空间开销。

本章工作通过强化学习框架赋予了微观传播模型预测宏观性质（即级联规模）

① 本章主要工作以 “Multi-scale Information Diffusion Prediction with Reinforced Recurrent Networks’’ 为题发表在 2019 年的国际学术会议 “The International Joint Conference on Artificial Intelligence (IJCAI’19)’’ 上。

(a) 微观下一个受影响用户预测　　(b) 宏观受影响用户规模预测

图 6.1　微观和宏观层面传播预测的示意图

的能力，从而提出了一种新颖的多层面传播预测模型。本工作进一步采用了一种快速且有效的结构上下文提取方法，该方法最初用于半监督图分类任务[157]。实验结果表明，本章工作提出的模型在微观和宏观传播预测任务上分别比最先进的基线方法取得了 10% 和 12% 的相对提升。

总结来说，本章工作有以下三个主要的贡献：

(1) 通过联合考虑微观和宏观传播预测任务，创新地提出了多层面传播预测问题。

(2) 本章工作提出使用强化学习框架来赋予微观传播模型预测宏观级联规模的能力，并采用了一种新颖的结构上下文提取算法，以进一步利用潜在的社交网络信息。

(3) 实验结果表明，本章工作提出的模型在三个真实世界数据集的微观和宏观传播预测效果都优于最先进的基线方法。

6.2　相关工作

本章工作主要与基于深度学习技术的微观和宏观传播预测算法相关。我们进一步根据它们的模型分为基于嵌入表示的和基于循环神经网络的方法。表6.1总结了相关工作的对比。这里 ‘‘历史信息’’ 和 ‘‘网络信息’’ 分别表示一个方法是否使用了受影响用户的激活顺序和社交网络信息。

6.2.1　基于嵌入表示的方法

基于嵌入表示的方法扩展了假设每个用户对间拥有独立传播概率的 IC 模型[112]，旨在进行微观传播预测。一些工作[124,125] 通过假设受影响的用户仅由源用户决定，将用户映射为实值用户向量并简化了 IC 模型。Embedded IC[123] 通过用户表示的函数来建模两个用户之间的传播概率。Inf2vec[158] 进一步考虑了全局用户

表 6.1　相关工作总结

方法	历史信息	社交网络	微观预测	宏观预测
Embedded IC			✓	
Inf2vec		✓	✓	
DeepCas	✓	✓		✓
DeepHawkes	✓			✓
CYAN-RNN	✓		✓	
TopoLSTM	✓	✓	✓	
DeepDiffuse	✓		✓	
NDM	✓	✓	✓	
SNIDSA	✓	✓	✓	
本章工作	✓	✓	✓	✓

相似性上下文作为扩展。但是，基于嵌入表示的方法没有在下一个受影响用户的预测中考虑传播历史，即受影响用户的顺序。近期基于循环神经网络的方法[128,156,159]的实验表明，基于嵌入表示的方法并不是最佳选择。

6.2.2　基于循环神经网络的方法

对于宏观传播预测模型，DeepCas[126] 从社交网络和级联中采样序列，然后用 RNN 编码序列并预测级联的最终规模。DeepHawkes[135] 在 RNN 框架基础上探索了 Hawkes 自激点过程来利用用户被影响的时间戳信息。

对于微观传播预测模型，TopoLSTM[128] 将循环神经网络中隐状态的序列结构替换为根据社交网络结构抽取的有向无环图。CYAN-RNN[160]，DAN[161] 和 Deep-Diffuse[155] 都使用了循环神经网络和注意力机制来利用用户被影响的时间戳信息。NDM[159] 基于两个启发式的假设，用自注意力机制和卷积神经网络构建微观传播模型。SNIDSA[156] 计算了所有用户两两之间的相似度并将结构信息通过门机制引入循环神经网络。然而据我们所知，还没有工作提出过能同时处理微观和宏观层面的统一模型。

6.3　模型框架

在本节中，我们首先形式化微观层面和宏观层面的传播预测问题。然后，我们将提出一种结构上下文提取算法以构建基于循环神经网络的微观传播模型。我们进一步通过强化学习框架将预测级联最终规模的能力引入模型。最后，我们将介绍整体算法和实现细节。

6.3.1　问题定义

给定用户集合 V 和级联集合 C，每个级联 $c_i \in C$ 是按受影响时间排序的用户序列 $\{v_1^i, v_2^i \ldots, v_{|c_i|}^i\}$，其中 $|c_i|$ 是级联 c_i 的规模，即对应传播对象影响的用户总数。在本章中，我们和先前工作[128,156,159] 一样只保留了用户被影响的顺序而忽略了他们被影响的具体时间。此外，当信息传播发生在社交网络服务上时，社交网络结构 $G = (V, E)$ 也可以被观测到。社交网络结构 G 将作为传播预测的额外结构输入。

本工作中，我们同时考虑细粒度的短期建模的微观传播预测和粗粒度的长期估计的宏观传播预测。我们将两个任务形式化如下：

微观层面的传播预测旨在给定级联 c_i 中已经被影响的用户 $\{v_1^i, v_2^i \ldots, v_k^i\}$ 的条件下，预测下一个被影响用户 v_{k+1}^i，其中 $k = 1, 2 \ldots, |c_i| - 1$。

宏观层面的传播预测旨在给定前 K 个被影响用户 $\{v_1^i, v_2^i \ldots, v_K^i\}$ 的条件下，预测级联 c_i 的最终规模 $|c_i|$。

6.3.2　微观传播建模

在这一小节中，我们将先介绍循环神经网络（RNN）的变体带门循环单元（GRU）作为微观层面建模的基础。然后我们介绍结构上下文提取算法来利用社交网络信息。之后，我们将展示如何将两者结合起来进行微观传播预测。

6.3.2.1　循环神经网络

循环神经网络已经在很多领域的序列数据建模中展示了有效性，例如自然语言处理[70]。将循环神经网络用于级联建模的先前工作[128,155,156] 都取得了相当不错的效果。特别地，我们使用带门循环单元（GRU）作为我们模型的基础。下面我们将简单介绍 GRU。

给定级联序列 $\{v_1, v_2 \ldots, v_k\}$，GRU 在每一步 $t = 1, 2 \ldots, k$ 中，将用户 v_t 作为输入并计算隐状态 h_t。公式（6-1）展示了第 t 步中隐状态 h_t 的计算过程。

$$
\begin{aligned}
r_t &= \sigma(W_{ir} x_{v_t} + b_{ir} + W_{hr} h_{t-1} + b_{hr}), \\
z_t &= \sigma(W_{iz} x_{v_t} + b_{iz} + W_{hz} h_{t-1} + b_{hz}), \\
n_t &= \tanh(W_{in} x_{v_t} + b_{in} + r_t * (W_{hn} h_{t-1} + b_{hn})), \\
h_t &= (1 - z_t) * n_t + z_t * h_{t-1},
\end{aligned}
\tag{6-1}
$$

其中 $x_{v_t} \in \mathbb{R}^d$ 是用户 v_t 的 d 维用户表示，$W \in \mathbb{R}^{d \times d}$ 和 $b \in \mathbb{R}^d$ 是权重矩阵和偏置向量。r_t, z_t, n_t 分别是重置门，更新门和新状态。σ 是 sigmoid 函数，$*$ 表示按位乘

积操作。

隐状态 $h_k \in \mathbb{R}^d$ 编码了级联中所有已经被影响的用户 $\{v_1, v_2 \ldots, v_k\}$ 的历史信息。现在我们进一步编码结构信息以利用用户间的社交网络。

6.3.2.2　结构上下文提取

对于每个用户 v，我们假设 $f_v^{(0)}$ 是其用户特征，可以从用户资料或者预训练的网络表示中得到。我们的目标是将结构信息引入用户特征 $f_v^{(0)}$ 中。受半监督图学习相关工作[157,162] 的启发，我们使用了基于邻居采样的高效结构上下文提取算法。

形式化地，我们首先从 v 和其邻居 $N(v)$ 中采样 Z 个用户 $\{u_1, u_2 \ldots, u_Z\}$。然后我们根据公式（6-2）聚合邻居特征来更新特征向量 $f_v^{(0)}$。

$$f_v^{(1)} = relu(W \cdot \frac{1}{Z} \sum_{k=1}^{Z} f_{u_k}^{(0)} + b), \tag{6-2}$$

其中对 $k = 1, 2 \ldots Z$，u_k 是从用户集合 $\{v\} \cup N(v)$ 中均匀采样的。W, b 是权重矩阵和偏置向量，激活函数 $relu(\cdot) = max(\cdot, 0)$。

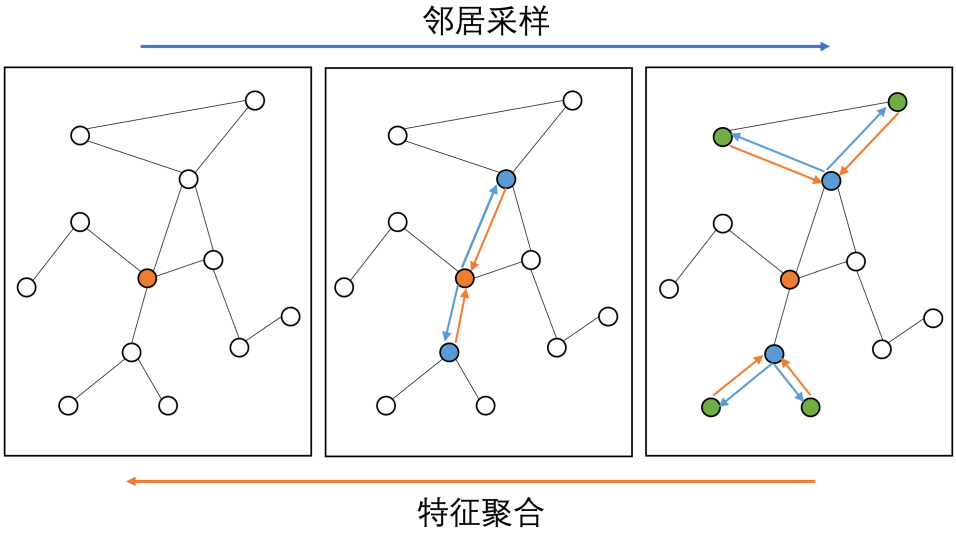


图 6.2　图中橙色节点的结构上下文提取示意图。首先从左至右进行邻居采样，然后从右至左进行特征聚合。

更新后的用户特征 $f_v^{(1)}$ 可以通过聚合来自 v 的一阶邻居的特征来编码结构信息。如图6.2所示，我们可以递归地使用公式（6-2）来探索用户 v 更大规模的邻域。经验上讲，一个两步的邻域探索可以快速地给出不错的效果。我们用 f_v 来表示最终的用户特征表示。

虽然最初用于半监督图分类任务，我们发现这个算法非常适用于我们的问题。和使用了 $O(|V|^2)$ 时间和空间复杂度进行结构上下文提取的先前工作[156] 相比，基于邻居采样和特征聚合的策略只使用了 $O(|V|)$ 的时间空间复杂度。此外，该算法可以在直接相连的邻居以外探索两步的邻域。

6.3.2.3　微观层面的传播预测

现在我们通过组合 GRU 和结构上下文来解决下一个受影响的用户预测问题。给定级联 c_i 中已经被影响的用户 $\{v_1^i, v_2^i \ldots, v_k^i\}$，GRU 第 k 步的隐状态 h_k^i 编码了序列历史，而用户特征 $f_{v_1^i}, f_{v_2^i}, \ldots f_{v_k^i}$ 编码了社交网络结构。

直觉上看，最近受影响的用户的结构上下文应该有助于下一个受影响用户的预测，因为信息很可能是通过用户间的社交链接传播的。因为我们忽略了具体的时间戳信息，我们定义 ‘‘最近受影响的用户’’ 为最近 m 个用户 $\{v_{k-m+1}^i, v_{k-m+2}^i \ldots, v_k^i\}$，其中 m 是控制这个窗口大小的超参数。我们进一步使用平均池化[①]来聚合用户特征为 $s_k^i = mean(f_{v_{k-m+1}^i}, f_{v_{k-m+2}^i}, \ldots f_{v_k^i})$。

最后，下个受影响用户的概率为

$$p_k^i = softmax(W_p \cdot concat(h_k^i, s_k^i) + b_p), \tag{6-3}$$

其中 $p_k^i \in \mathbb{R}^{|V|}$ 是所有用户上的多项式概率分布，$concat(\cdot,\cdot)$ 是拼接操作，W_p, b_p 分别是权重矩阵和偏置向量。

微观传播预测的训练目标为最大化所有级联的对数似然

$$J_{micro}(\Theta) = \sum_{i=1}^{|C|} \sum_{k=1}^{|c_i|-1} \log p_k^i[v_{k+1}^i], \tag{6-4}$$

其中 $p[j]$ 表示向量 p 的第 j 维，Θ 表示微观传播模型中的所有参数。

6.3.3　宏观传播建模

本章工作的关键在于如何赋予微观级联模型预测宏观级联规模的能力。我们将方法分为四个步骤：(a) 用微观级联模型编码观察到的 K 个用户；(b) 通过模拟级联的生成过程使得微观级联模型能够预测级联的规模；(c) 使用 Mean-Square Log-Transformed Error (MSLE) 作为宏观预测的监督信号；(d) 建立强化学习框架并利用 policy gradient 算法更新参数。整个流程如图6.3所示。

① 我们也尝试了诸如注意力机制或拼接的其他聚合策略。但其他选择会导致模型过拟合或次优的效果。

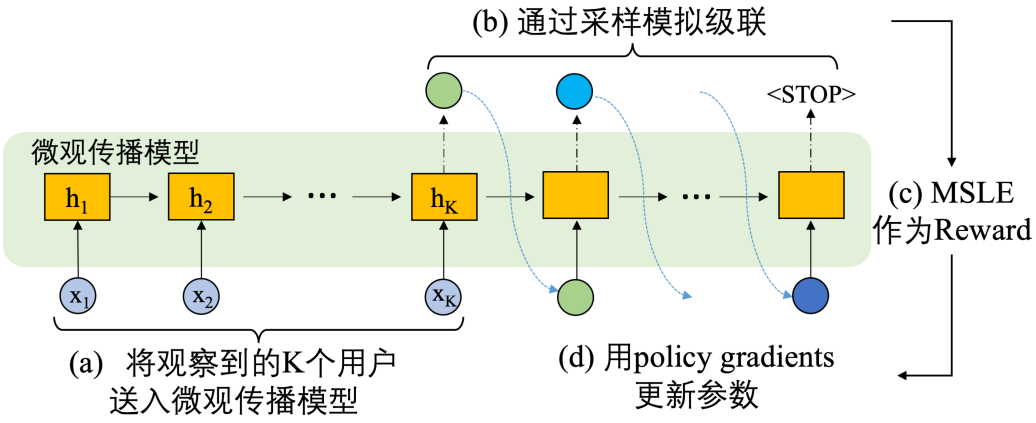


图 6.3　通过强化学习使微观级联模型能够进行宏观级联规模预测的流程图。

6.3.3.1　编码观察到的受影响用户

如图6.3中步骤 (a) 所示，的我们将级联 c_i 中观察到的 K 个用户送入微观级联模型，并得到最后的隐状态 h_K^i。另外，我们也在每一步中显式地编码位置信息，使得模型知道有多少用户已经被输入到 GRU 中。具体地，我们为每步 $t = 1, 2 \ldots maxlen$ 分配一个位置表示 $POS_t \in \mathbb{R}^{d_{pos}}$，其中 $maxlen$ 是预设的级联最大长度。在公式（6-1）中 GRU 的第 t 步，我们将用户表示 x_{v_t} 和位置表示 POS_t 拼接作为输入向量。

6.3.3.2　用于宏观预测的级联模拟

为了在步骤 (b) 中使得微观级联模型能够进行级联规模预测，我们首先在每个级联末尾添加虚拟用户 <STOP> 并让模型也将其看作普通用户进行预测。给定前 K 个受影响用户，为了估算整个级联的规模，我们递归地根据公式（6-3）中预测的概率分布采样一个用户，将其作为下一步的输入并继续预测。一旦我们采样到 <STOP> 信号，我们认为该级联的传播停止并统计已经被预测的用户数作为级联的最终规模。级联模拟过程将被重复多次以减小估算的方差。

6.3.3.3　宏观预测的监督信号

虽然修改后的微观级联模型能够通过模拟预测级联的规模，但是模型仍然没有任何监督信号来引导模型获得更好的效果。在本章工作中，我们使用先前工作[126,135] 中使用的 Mean Square Log-transformed Error (MSLE) 作为级联规模预测的评价标准，亦即图 6.3步骤 (c) 中的监督信号：MSLE= $\frac{1}{|C|}\sum_{i=1}^{|C|}(\log(|c_i|)-\log(pred_i))^2$ 其中 $pred_i$ 是级联 c_i 的预测规模。

然而，级联规模估计中使用的采样操作是不可导的，使得无法通过后向传播

更新参数。为了解决这个问题，如图6.3的步骤 (d) 所示，我们将模拟过程放入强化学习的框架中并使用 policy gradient 算法更新参数。

6.3.3.4　用于参数更新的 Policy gradient

我们将 GRU 及其隐状态 (包括结构上下文) 对应到强化学习中的 *agent* 和 *state* 概念。每一步的 *action* 是选取下个受影响用户，而给定当前 state 决定每个 action 概率的 *policy* 由公式（6-3）决定。当 <STOP> 这个特殊的 action 被选中时，我们将 MSLE 的相反数[①]作为 *reward* 给出。

形式化地，为了预测级联 c_i 的规模，c_i 的前 K 个用户被送入微观级联模型且最后一个隐状态 h_K^i 被用作强化学习的初始状态。对于每个 action 序列 $seq = \{a_1, a_2 \dots a_{maxlen}\}$ 其中 a_j 是第 j 个 action 选择的用户，我们可以计算 MSLE 的相反数作为 $reward(seq, c_i)$。然后我们的目标是最大化级联 c_i 的 reward 的期望

$$J_{RL}^i(\Theta) = \sum_{seq} \Pr(seq; \Theta, h_K^i) reward(seq, c_i), \tag{6-5}$$

其中 $\Pr(seq; \Theta, h_K^i)$ 是选取 action 序列 seq 的概率，可以被分解为每个 action a_j 的概率的乘积。注意 seq 所处的集合空间是 $|V|^{maxlen}$，无法枚举计算 J_{RL}^i。取而代之的是，我们用 REINFORCE 算法[163] 来计算 J_{RL}^i 的梯度：

$$\begin{aligned}
\nabla_\Theta J_{RL}^i &= \sum_{seq} \nabla_\Theta \Pr(seq; \Theta, h_K^i) \cdot reward(seq, c_i) \\
&= \sum_{seq} \Pr(seq; \Theta, h_K^i) \nabla_\Theta \log \Pr(seq; \Theta, h_K^i) \cdot reward(seq, c_i) \\
&= \mathbb{E}_{seq}[\nabla_\Theta \log \Pr(seq; \Theta, h_K^i) \cdot reward(seq, c_i)] \\
&\simeq \frac{1}{M} \sum_{m=1}^{M} \nabla_\Theta \log \Pr(seq_m; \Theta, h_K^i) \cdot reward(seq_m, c_i),
\end{aligned} \tag{6-6}$$

其中 $seq_m, m = 1, 2 \dots M$ 是从 $\Pr(seq; \Theta, h_K^i)$ 中随机抽取的 M 个采样，最后一步中整个 action 序列的期望由蒙特卡洛模拟近似。最后，参数 Θ 通过最大化 reward（即宏观预测的监督信号）的期望的梯度上升来更新。

① 如果 <STOP> 当级联到达最大长度 $maxlen$ 时也没有出现，我们假设这次模拟预测了 $2 \times maxlen$ 的级联规模。

6.3.4　实现细节

为了进行微观和宏观训练目标的联合训练，我们首先训练 10 轮微观级联模型作为初始化。该初始化过程可以保证下个受影响用户的预测概率 p_k^i 可以模拟出高质量的级联，从而帮助强化学习的训练更快地收敛。之后我们迭代地最大化微观和宏观目标来更新参数，直到我们在验证集上到达最大值。我们用 Adam[149] 来进行梯度上升。

在超参数设置上，隐状态和用户特征表示的维度 $d = 64$，窗口大小 $m = 3$，结构上下文提取中的一阶和二阶的采样邻居数 $Z_1 = 25, Z_2 = 10$，位置表示的维度 $d_{pos} = 8$，batch 大小为 16。

我们将模型命名为 reinFOrced REcurrent networks with STructural context (FOREST)。

6.4　实验结果

我们在微观和宏观级联预测任务上进行实验来证明我们提出模型的有效性。我们首先介绍数据集，基线方法和实验设置。然后展示讨论实验结果。

6.4.1　数据集

我们调研了微观传播预测的前人工作[128,155,159] 中使用的数据集，并去掉了拥有固定或者过小的级联规模的数据集。我们使用了三个真实世界数据集 Twitter，Douban 和 Memetracker 进行实验。

Twitter[142] 数据集记录了 2010 年 10 月期间包含 URL 链接的微博信息。每个 URL 被看作一个用户间传播的信息。

Douban[164] 是用户可以更新其读书状态并追踪他人状态的社交网站。每本书被看作一个传播对象。

Memetracker[117] 收集了一百万条新闻报道和博客文章，并追踪其中最常见的引用和短语，即模因，来研究模因在人群中的传播。每个模因被看作一个传播对象而每个网站或博客的 URL 则被看作一个用户。注意对于这个数据集我们没有社交网络结构信息。

我们随机采样了 80% 的级联用于训练，10% 作为验证集，其余的 10% 作为测试。数据集的统计信息如表6.2所示。

6.4.2　基线方法

为了进行全面的比较，我们考虑了五个微观和宏观传播预测的基线方法。

表 6.2　数据集统计信息

数据集	# 用户	# 链接	# 级联	平均长度
Twitter	12,627	309,631	3,442	32.60
Douban	23,123	348,280	10,602	27.14
Memetracker	4,709	-	12,661	16.24

微观传播预测模型：

TopoLSTM[128] 将循环神经网络中隐状态的序列结构替换为根据社交网络结构抽取的有向无环图进行微观层面的预测。

DeepDiffuse[155] 利用表示学习技术和注意力机制来利用受影响时间戳信息。

NDM[159] 基于自注意力机制和卷积神经网络来构建微观级联模型，并缓解了远距离依赖的问题。

SNIDSA[156] 计算了所有用户对之间的相似度并将结构信息通过门机制引入循环神经网络中。

宏观传播预测模型：

DeepCas[126] 是考虑了级联信息和社交网络的最先进的级联规模预测算法。

6.4.3　实验设置

对于微观层面预测，我们通过对所有未被影响用户按公式（6-3）中的被影响概率进行排序，将下个受影响用户预测看作一个抽取任务。我们报告了 Mean Average Precision (MAP) 和 HITS 值。同样的设置也被用于前人工作[128,155] 中。

对于宏观层面预测，我们给出级联中前 $K = 5$ 个用户来预测整个级联的规模。和前人工作[126] 一致，我们用6.3.3.3小节中介绍的 MSLE 作为评价指标。另外，所有微观层面的基线方法都通过在训练级联末尾添加额外的 <STOP> 信号的方式用于宏观级联规模预测。

对 Twitter 和 Douban 数据集，我们用 $d = 64$ 维的预训练的 DeepWalk[154] 表示作为初始的用户特征 $f_v^{(0)}$。我们在 Memetracker 数据集上忽略了 TopoLSTM 和 SNIDSA 方法，因为该数据集没有社交网络信息。

6.4.4　传播预测实验结果与分析

图6.4和6.5分别展示了微观和宏观传播预测任务的实验结果。我们有如下观察：

(1) FOREST 在微观传播预测任务上一致地超过所有基线方法，并在 HITS 和 MAP 指标上取得了超过 10% 的相对提升。与 TopoLSTM 和 SNIDSA 相比，提升主要来自于结构上下文的编码。FOREST 的结构上下文编码考虑了二阶邻居而先

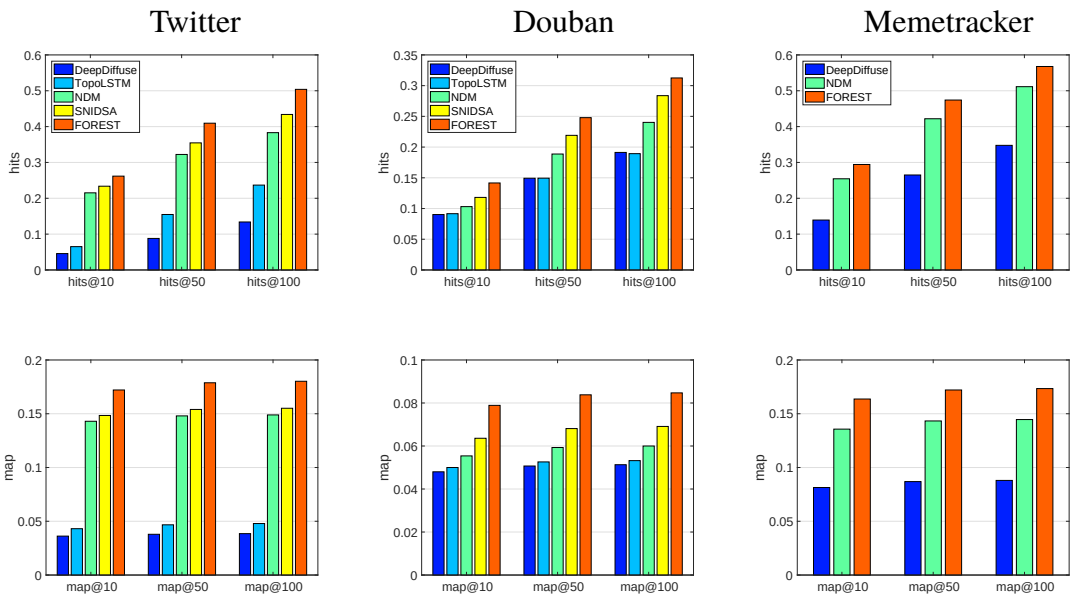


图 6.4　微观传播预测任务实验结果。MAP 和 HITS 都是越高越好。

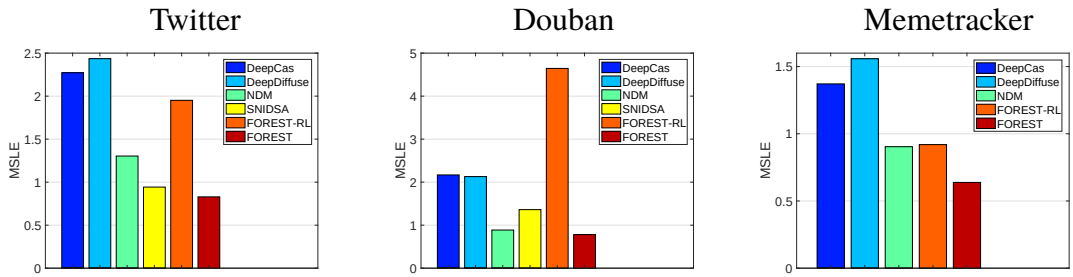


图 6.5　宏观传播预测任务实验结果。MSLE 评价指标越小越好。FOREST-RL 我们提出的 FOREST 方法去掉强化学习训练部分的变体。我们省略了 TopoLSTM 的实验结果因为其 MSLE 大于 10，无法放到图中。

前工作都只考虑了一阶邻居。此外，如6.3.2.2节中所介绍，FOREST 使用的计算资源远远少于 SNIDSA。

(2) FOREST 在宏观传播预测任务上一致地超过包含最先进的级联规模预测算法 DeepCas 在内的所有基线方法，并在 MSLE 评价指标上取得了 12% 的相对提升。和去掉了强化学习模块的 FOREST-RL 相比，FOREST 通过引入宏观监督信号来学习参数，取得了更显著且鲁棒的效果。

(3) 和 DeepCas 相比，微观级联模型能够利用训练数据中更多的信息，即具体的受影响用户和他们的受影响顺序。因此微观级联模型能够在宏观预测任务上给出相当甚至更好的实验效果。这一发现将鼓励微观级联模型在将来的工作中取代宏观模型。

(4) 总体上讲，微观级联模型给出了相当不错的性能。以 Douban 数据集为例，当我们从全部 23,123 个用户中预测 10 个用户时，我们有接近 15% 的概率预测出真实的被影响用户。这一结果表明，从传播预测的角度的相关工作也可能在将来服务于实时广告任务。

6.5　本章小结

在本章中，我们提出了一个新颖的多层面传播预测模型 FOREST，能够同时进行微观和宏观层面的传播预测。具体地，我们通过强化学习框架将宏观预测的监督信号引入微观级联模型，并取得了相当优异的结果。另外，我们使用了快速有效的结构上下文提取方法来利用社交网络结构信息。下个受影响用户预测和级联规模预测任务上的实验结果证明了我们提出的方法的有效性。

第 7 章　总结与展望

网络是表达对象与对象间关系的常用数据形式。除去网络的拓扑结构信息之外，真实的网络数据中一般还包含着根据节点的属性、行为等产生的丰富信息，统称为富信息网络。随着互联网技术和移动智能设备的发展，富信息网络数据的规模飞速增长，并带来了丰富的应用任务和巨大的市场价值。如何让已经在多个领域取得巨大成功的机器学习，特别是深度学习技术，服务于富信息网络数据及其典型应用已经在近年来成为人工智能领域的研究热点。传统的邻接矩阵形式的网络表示具有维度过高和数据稀疏两大缺点，使得研究者们无法在网络数据上应用机器学习和深度学习技术。因此，现有工作通过网络表示学习为网络中每个节点学习一个实数向量表示以编码其拓扑结构。节点的向量表示可看作其特征，送入支持向量机（SVM）等机器学习分类器用于节点分类、链接预测等任务。本文针对现有的富信息网络表示学习工作中**缺乏对于已有网络表示学习算法的理论分析**、**忽略了网络拓扑结构以外的丰富信息**和**难以应用于相对复杂的典型应用问题**三个方面的问题，从富信息网络表示学习和典型应用问题研究两个层面，系统性地进行了五个工作，在多个富信息网络应用问题中取得了远超基线方法的结果。

7.1　主要贡献

本文的主要贡献包括以下三点：

- 针对缺乏对于已有网络表示学习算法的理论分析的问题，本文提出了**网络表示学习的统一框架和增强算法**。本工作将大多数现有的只考虑拓扑结构信息的一般网络表示学习方法总结为一个统一的两步框架：邻近度矩阵构造和降维。本工作侧重于邻近度矩阵构造步骤的分析，并得出结论，如果在构建邻近度矩阵时探索了更高阶的邻近度，网络表示学习算法可以得到增强。本工作进一步提出了网络嵌入更新（NEU）算法，该算法从理论上隐含地近似了高阶邻近度，并且可以应用于任何网络表示学习方法以提高它们的性能。多标签分类和链接预测任务上的实验结果表明，NEU 在所有三个公开可用的数据集上可以对许多网络表示学习方法进行一致且显著的增强，且运行时间几乎可以忽略不计。
- 针对现有网络表示学习方法忽略了网络拓扑结构以外的丰富信息的问题，本文提出了**结合富特征信息的网络表示学习**。受前一工作中得到的最先进的网络表示学习算法 DeepWalk[1] 实际上等同于一种特殊的矩阵分解的结论的启

发，该工作以文本特征为例，在矩阵分解的框架下将节点的特征结合到网络表示学习中，并将学习得到的节点表示应用于节点多分类任务来评估本工作的方法和各种基线方法。实验结果表明，该方法在所有三个数据集上均优于其他基线方法，特别是当网络结构噪声较大或者训练数据比例较小时。

- 针对现有网络表示学习方法难以应用于相对复杂的典型应用问题的缺点，本文以网络表示学习技术作为模型的底层，并根据特定的富信息网络场景利用包括循环神经网络、卷积神经网络在内的深度学习模型进行建模，在推荐系统和传播预测两个典型的富信息网络应用问题中，创新性地提出了**基于位置的社交网络的推荐系统**、**微观层面的信息传播预测**和**多层面的信息传播预测**的应用模型。多个真实数据集上的实验结果证明了所提出的方法的有效性和鲁棒性。

7.2　未来工作展望

富信息网络表示学习及其相关应用领域的一大研究趋势是富信息网络场景的逐渐细化，即在特定网络场景或算法需求下设计模型。潜在的未来研究方向包括：

- **超大规模的富信息网络表示学习**：现有网络表示学习算法可以应用于百万级规模的网络数据。然而在实际的工业界场景中，网络中的节点数和链接数很可能有上亿规模，单单是网络表示的存储就会花费很大的计算开销。如何利用并行化计算、分布式存储等技术手段进行超大规模的网络表示学习，对于工业界实际应用的开发具有十分重要的意义。
- **结合更多信息的富信息网络表示应用**：虽然本文提出的用于推荐系统和传播预测的相关模型针对特定场景进行了设计，但在建模时也有一定程度的简化。如第4章基于位置的社交网络的推荐系统中忽略了位置的 GPS 信息；在第5章和第6章的传播预测模型中忽略了用户被影响时具体的时间戳信息。如何在富信息网络应用模型中结合更多的信息，是该领域的另一研究方向。
- **基于表示学习的隐式网络结构推断**：网络结构数据是可解释性强、易于理解和可视化的数据形式。但是在一些场景中，网络结构是无法直接获取的，例如文本语料库中不同文字风格的混淆度关系[165] 或者不同实体之间的关系网络[166]。通过表示学习对隐含的网络结构进行推断，有助于对数据中的规律模式进行可视化并更好地理解数据。

总之，富信息网络表示学习及典型应用问题具有非常广阔的研究前景，对上述未来研究方向的工作将进一步充实该领域下的研究成果，并促成相关技术在工业界实际任务中的落地应用，具有极高的研究价值。

参考文献

[1] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations[C]// Proceedings of SIGKDD. 2014: 701–710.

[2] Tang L, Liu H. Leveraging social media networks for classification[J]. Proceedings of SIGKDD, 2011, 23(3): 447–478.

[3] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Proceedings of NIPS. 2013: 3111–3119.

[4] Grover A, Leskovec J. node2vec: Scalable feature learning for networks[M]//Proceedings of KDD. 2016.

[5] Tang J, Qu M, Wang M, et al. Line: Large-scale information network embedding[C]// Proceedings of WWW. 2015.

[6] Cao S, Lu W, Xu Q. Grarep: Learning graph representations with global structural information [C]//Proceedings of CIKM. 2015.

[7] Ley M. The dblp computer science bibliography: Evolution, research issues, perspectives[C]// International symposium on string processing and information retrieval. 2002.

[8] Sen P, Namata G M, Bilgic M, et al. Collective classification in network data[J]. AI Magazine, 2008.

[9] Tu C, Liu Z, Sun M. Inferring correspondences from multiple sources for microblog user tags [C]//Proceedings of SMP. 2014: 1–12.

[10] Akoglu L, Tong H, Koutra D. Graph based anomaly detection and description: a survey[J]. Data Mining and Knowledge Discovery, 2015.

[11] Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks[J]. journal of the Association for Information Science and Technology, 2007.

[12] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives[J]. IEEE PAMI, 2013, 35(8): 1798–1828.

[13] Tu C, Zhang W, Liu Z, et al. Max-margin deepwalk: discriminative learning of network representation[C]//Proceedings of IJCAI. 2016.

[14] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[C]// Proceedings of ICLR. 2017.

[15] Li J, Zhu J, Zhang B. Discriminative deep random walk for network classification[C]// Proceedings of ACL. 2016.

[16] Yang Z, Cohen W, Salakhutdinov R. Revisiting semi-supervised learning with graph embeddings [C]//Proceedings of ICML. 2016.

[17] Wang D, Cui P, Zhu W. Structural deep network embedding[C]//Proceedings of KDD. 2016.

[18] Ou M, Cui P, Pei J, et al. Asymmetric transitivity preserving graph embedding[C]//Proceedings of KDD. 2016.

[19] Wang X, Cui P, Wang J, et al. Community preserving network embedding[J]. Proceedings of AAAI, 2017.

[20] Tang J, Qu M, Mei Q. Pte: Predictive text embedding through large-scale heterogeneous text networks[C]//Proceedings of KDD. 2015.

[21] Chang S, Han W, Tang J, et al. Heterogeneous network embedding via deep architectures[C]// Proceedings of KDD. 2015.

[22] Huang Z, Mamoulis N. Heterogeneous information network embedding for meta path based proximity[J]. arXiv preprint arXiv:1701.05291, 2017.

[23] Xu L, Wei X, Cao J, et al. Embedding of embedding (eoe): Joint embedding for coupled heterogeneous networks[C]//Proceedings of WSDM. 2017.

[24] Huang X, Li J, Hu X. Label informed attributed network embedding[C]//Proceedings of WSDM. 2017.

[25] Fouss F, Pirotte A, Renders J M, et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation[J]. IEEE TKDE, 2007, 19(3): 355–369.

[26] Morin F, Bengio Y. Hierarchical probabilistic neural network language model[C]//Proceedings of AISTATS: volume 5. 2005: 246–252.

[27] Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization[C]//Proceedings of NIPS. 2014: 2177–2185.

[28] Yang C, Liu Z, Zhao D, et al. Network representation learning with rich text information.[C]// Proceedings of IJCAI. 2015.

[29] Tang L, Liu H. Relational learning via latent social dimensions[C]//Proceedings of SIGKDD. 2009: 817–826.

[30] Fan R E, Chang K W, Hsieh C J, et al. Liblinear: A library for large linear classification[J]. JMLR, 2008, 9: 1871–1874.

[31] Ben-Hur A, Weston J. A userąfs guide to support vector machines[J]. Data mining techniques for the life sciences, 2010.

[32] Hanley J A, McNeil B J. The meaning and use of the area under a receiver operating characteristic (roc) curve.[J]. Radiology, 1982.

[33] Salton G, McGill M J. Introduction to modern information retrieval[M]. : McGraw-Hill, Inc., 1986.

[34] Sen P, Namata G, Bilgic M, et al. Collective classification in network data[J]. AI magazine, 2008, 29(3): 93.

[35] Bhuyan M H, Bhattacharyya D, Kalita J K. Network anomaly detection: methods, systems and tools[J]. IEEE Communications Surveys & Tutorials, 2014, 16(1): 303–336.

[36] Lü L, Zhou T. Link prediction in complex networks: A survey[J]. Physica A: Statistical Mechanics and its Applications, 2011, 390(6): 1150–1170.

[37] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Proceedings of NIPS. 2012: 1097–1105.

[38] Chen M, Yang Q, Tang X. Directed graph embedding.[C]//Proceedings of IJCAI. 2007: 2707–2712.

[39] Mei Q, Cai D, Zhang D, et al. Topic modeling with network regularization[C]//Proceedings of WWW. 2008: 101–110.

[40] Yu H F, Jain P, Dhillon I S. Large-scale multi-label learning with missing labels[C]//Proceedings of ICML. 2014.

[41] Natarajan N, Dhillon I S. Inductive matrix completion for predicting gene–disease associations [J]. Bioinformatics, 2014, 30(12): i60–i68.

[42] Hofmann T. Probabilistic latent semantic indexing[C]//Proceedings of SIGIR. 1999: 50–57.

[43] Joachims T. Making large-scale SVM learning practical[M]//Schölkopf B, Burges C, Smola A. Advances in Kernel Methods - Support Vector Learning. Cambridge, MA: MIT Press, 1999: 169–184

[44] Shi X, Li Y, Yu P. Collective prediction with latent graphs[C]//Proceedings of CIKM. 2011: 1127–1136.

[45] McDowell L, Aha D. Semi-supervised collective classification via hybrid label regularization [C]//Proceedings of ICML. 2012.

[46] McDowell L K, Aha D W. Labels or attributes?: rethinking the neighbors for collective classification in sparsely-labeled networks[C]//Proceedings of CIKM. 2013: 847–852.

[47] Cho Y S, Ver Steeg G, Galstyan A. Socially relevant venue clustering from check-in data[C]// KDD Workshop on Mining and Learning with Graphs. 2013.

[48] Cho E, Myers S A, Leskovec J. Friendship and mobility: user movement in location-based social networks[C]//Proceedings of SIGKDD. 2011.

[49] Bao J, Zheng Y, Mokbel M F. Location-based and preference-aware recommendation using sparse geo-social networking data[C]//Proceedings of the 20th international conference on advances in geographic information systems. : ACM, 2012: 199–208.

[50] Zheng Y. Trajectory data mining: an overview[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2015, 6(3): 29.

[51] Machanavajjhala A, Korolova A, Sarma A D. Personalized social recommendations: accurate or private[J]. Proceedings of the VLDB Endowment, 2011, 4(7): 440–450.

[52] Yuan Q, Cong G, Lin C Y. Com: a generative model for group recommendation[C]//Proceedings of SIGKDD. : ACM, 2014: 163–172.

[53] Ma H. On measuring social friend interest similarities in recommender systems[C]//Proceedings of SIGIR. : ACM, 2014: 465–474.

[54] Pham H, Shahabi C, Liu Y. Ebm: an entropy-based model to infer social strength from spatiotemporal data[C]//Proceedings of SIGMOD. : ACM, 2013: 265–276.

[55] Zheng Y, Xie X, Ma W Y. Geolife: A collaborative social networking service among user, location and trajectory.[J]. IEEE Data Eng. Bull., 2010, 33(2): 32–39.

[56] Zheng Y, Zhang L, Ma Z, et al. Recommending friends and locations based on individual location history[J]. ACM Transactions on the Web (TWEB), 2011, 5(1): 5.

[57] Chandola V, Banerjee A, Kumar V. Anomaly detection: A survey[J]. ACM computing surveys (CSUR), 2009.

[58] Fortunato S. Community detection in graphs[J]. Physics Reports, 2010.

[59] Zheng Y, Zhang L, Xie X, et al. Mining interesting locations and travel sequences from gps trajectories[C]//Proceedings of WWW. : ACM, 2009: 791–800.

[60] Cheng C, Yang H, King I, et al. Fused matrix factorization with geographical and social influence in location-based social networks.[C]//Proceedings of AAAI: volume 12. 2012: 17–23.

[61] Ye M, Yin P, Lee W C, et al. Exploiting geographical influence for collaborative point-of-interest recommendation[C]//Proceedings of SIGIR. : ACM, 2011: 325–334.

[62] Yuan Q, Cong G, Ma Z, et al. Time-aware point-of-interest recommendation[C]//Proceedings of SIGIR. : ACM, 2013: 363–372.

[63] Yuan Q, Cong G, Sun A. Graph-based point-of-interest recommendation with geographical and temporal influences[C]//Proceedings of CIKM. : ACM, 2014: 659–668.

[64] Liu Q, Wu S, Wang L, et al. Predicting the next location: A recurrent model with spatial and temporal contexts[C]//Proceedings of AAAI. 2016.

[65] Cheng C, Yang H, Lyu M R, et al. Where you like to go next: Successive point-of-interest recommendation[C]//Proceedings of IJCAI. 2013.

[66] Ye J, Zhu Z, Cheng H. Whatąŕs your next move: User activity prediction in location-based social networks[C]//Proceedings of the SIAM International Conference on Data Mining. SIAM. 2013.

[67] Zhang J D, Chow C Y, Li Y. Lore: Exploiting sequential influence for location recommendations [C]//Proceedings of SIGSPATIAL. : ACM, 2014: 103–112.

[68] Levandoski J J, Sarwat M, Eldawy A, et al. Lars: A location-aware recommender system[C]// 2012 IEEE 28th International Conference on Data Engineering. : IEEE, 2012: 450–461.

[69] Mittal S. A survey of techniques for approximate computing[J]. ACM Computing Surveys (CSUR), 2016, 48(4): 62.

[70] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model.[C]// Interspeech: volume 2. 2010: 3.

[71] Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. [C]//Proceedings of NIPS. 2001.

[72] Yan S, Xu D, Zhang B, et al. Graph embedding and extensions: a general framework for dimensionality reduction[J]. PAMI, 2007.

[73] Tu C, Zhang W, Liu Z, et al. Max-margin deepwalk: Discriminative learning of network representation[M]. 2016.

[74] Chung J, Gulcehre C, Cho K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.

[75] Zhang Y, Dai H, Xu C, et al. Sequential click prediction for sponsored search with recurrent neural networks[C]//Proceedings of AAAI. 2014.

[76] Romero D M, Kleinberg J M. The directed closure process in hybrid social-information networks, with an analysis of link formation on twitter.[C]//ICWSM. 2010.

[77] Huang H, Tang J, Wu S, et al. Mining triadic closure patterns in social networks[C]//Proceedings of WWW. : ACM, 2014: 499–504.

[78] Li Q, Zheng Y, Xie X, et al. Mining user similarity based on location history[C]//Proceedings of SIGSPATIAL. : ACM, 2008: 34.

[79] Xiao X, Zheng Y, Luo Q, et al. Finding similar users using category-based location history[C]// Proceedings of SIGSPATIAL. : ACM, 2010: 442–445.

[80] Cranshaw J, Toch E, Hong J, et al. Bridging the gap between physical location and online social networks[C]//Proceedings of the 12th ACM international conference on Ubiquitous computing. : ACM, 2010: 119–128.

[81] Lee M J, Chung C W. A user similarity calculation based on the location for social network services[C]//International Conference on Database Systems for Advanced Applications. : Springer, 2011: 38–52.

[82] Wang D, Pedreschi D, Song C, et al. Human mobility, social ties, and link prediction[C]// Proceedings of SIGKDD. : ACM, 2011: 1100–1108.

[83] Pham H, Hu L, Shahabi C. Towards integrating real-world spatiotemporal data with social networks[C]//Proceedings of SIGSPATIAL. : ACM, 2011: 453–457.

[84] Zhao W X, Zhou N, Zhang W, et al. A probabilistic lifestyle-based trajectory model for social strength inference from human trajectory data[J]. ACM Transactions on Information Systems (TOIS), 2016, 35(1): 8.

[85] Yin H, Sun Y, Cui B, et al. Lcars: a location-content-aware recommender system[C]// Proceedings of SIGKDD. : ACM, 2013: 221–229.

[86] Zhou N, Zhao W X, Zhang X, et al. A general multi-context embedding model for mining human trajectory data[J]. IEEE transactions on knowledge and data engineering, 2016.

[87] Gao H, Tang J, Hu X, et al. Content-aware point of interest recommendation on location-based social networks.[C]//Proceedings of AAAI. : Citeseer, 2015: 1721–1727.

[88] Li Y, Nie J, Zhang Y, et al. Contextual recommendation based on text mining[C]//Proceedings of the 23rd International Conference on Computational Linguistics: Posters. : Association for Computational Linguistics, 2010: 692–700.

[89] Zhao K, Cong G, Yuan Q, et al. Sar: a sentiment-aspect-region model for user preference analysis in geo-tagged reviews[C]//2015 IEEE 31st International Conference on Data Engineering. : IEEE, 2015: 675–686.

[90] Zheng V W, Zheng Y, Xie X, et al. Collaborative location and activity recommendations with gps history data[C]//Proceedings of WWW. : ACM, 2010: 1029–1038.

[91] Bhargava P, Phan T, Zhou J, et al. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data[C]//Proceedings of WWW. : ACM, 2015: 130–140.

[92] Gao H, Tang J, Hu X, et al. Exploring temporal effects for location recommendation on location-based social networks[C]//Proceedings of the 7th ACM conference on Recommender systems. : ACM, 2013: 93–100.

[93] Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized markov chains for next-basket recommendation[C]//Proceedings of WWW. 2010.

[94] Feng S, Li X, Zeng Y, et al. Personalized ranking metric embedding for next new poi recommendation[C]//Proceedings of IJCAI. 2015.

[95] Wang P, Guo J, Lan Y, et al. Learning hierarchical representation model for nextbasket recommendation[C]//Proceedings of SIGIR. 2015.

[96] Werbos P J. Backpropagation through time: what it does and how to do it[J]. Proceedings of the IEEE, 1990.

[97] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization[J]. The Journal of Machine Learning Research, 2011, 12: 2121–2159.

[98] Le Q V, Mikolov T. Distributed representations of sentences and documents[J]. Computer Science, 2014, 4: 1188–1196.

[99] Mnih A, Salakhutdinov R. Probabilistic matrix factorization[C]//Proceedings of NIPS. 2007.

[100] Tu C, Liu Z, Sun M. Prism: Profession identification in social media with personal information and community structure[C]//Chinese National Conference on Social Media Processing. : Springer, 2015: 15–27.

[101] Salganik M J, Dodds P S, Watts D J. Experimental study of inequality and unpredictability in an artificial cultural market[J]. science, 2006.

[102] Cheng J, Adamic L, Dow P A, et al. Can cascades be predicted?[C]//Proceedings of WWW. : ACM, 2014.

[103] Yu L, Cui P, Wang F, et al. From micro to macro: Uncovering and predicting information cascading process with behavioral dynamics[C]//Proceedings of ICDM. 2015.

[104] Zhao Q, Erdogdu M A, He H Y, et al. Seismic: A self-exciting point process model for predicting tweet popularity[C]//Proceedings of SIGKDD. 2015.

[105] Domingos P, Richardson M. Mining the network value of customers[C]//Proceedings of SIGKDD. : ACM, 2001.

[106] Leskovec J, Singh A, Kleinberg J. Patterns of influence in a recommendation network[C]// Pacific-Asia Conference on Knowledge Discovery and Data Mining. : Springer, 2006: 380–389.

[107] Leskovec J, Adamic L A, Huberman B A. The dynamics of viral marketing[J]. ACM Transactions on the Web (TWEB), 2007, 1(1): 5.

[108] Watts D J, Dodds P S. Influentials, networks, and public opinion formation[J]. Journal of consumer research, 2007.

[109] Aral S, Walker D. Identifying influential and susceptible members of social networks[J]. Science, 2012.

[110] Hethcote H W. The mathematics of infectious diseases[J]. SIAM review, 2000, 42(4): 599–653.

[111] Wallinga J, Teunis P. Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures[J]. American Journal of epidemiology, 2004.

[112] Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network [C]//Proceedings of SIGKDD. : ACM, 2003: 137–146.

[113] Lappas T, Terzi E, Gunopulos D, et al. Finding effectors in social networks[C]//Proceedings of SIGKDD. 2010.

[114] Dow P A, Adamic L A, Friggeri A. The anatomy of large facebook cascades.[J]. ICWSM, 2013.

[115] Gruhl D, Guha R, Liben-Nowell D, et al. Information diffusion through blogspace[C]// Proceedings of WWW. 2004.

[116] Liben-Nowell D, Kleinberg J. Tracing information flow on a global scale using internet chain-letter data[J]. Proceedings of the national academy of sciences, 2008, 105(12): 4633–4638.

[117] Leskovec J, Backstrom L, Kleinberg J. Meme-tracking and the dynamics of the news cycle[C]// Proceedings of SIGKDD. 2009.

[118] Romero D M, Meeder B, Kleinberg J. Differences in the mechanics of information diffusion across topics: idioms, political hashtags, and complex contagion on twitter[C]//Proceedings of WWW. : ACM, 2011: 695–704.

[119] Myers S, Leskovec J. On the convexity of latent social network inference[C]//Proceedings of NIPS. 2010.

[120] Gomez Rodriguez M, Leskovec J, Krause A. Inferring networks of diffusion and influence[C]// Proceedings of SIGKDD. : ACM, 2010.

[121] Rodriguez M G, Leskovec J, Balduzzi D, et al. Uncovering the structure and temporal dynamics of information propagation[J]. Network Science, 2014, 2(1): 26–65.

[122] Gomez Rodriguez M, Leskovec J, Schölkopf B. Structure and dynamics of information pathways in online media[C]//Proceedings of WSDM. : ACM, 2013.

[123] Bourigault S, Lamprier S, Gallinari P. Representation learning for information diffusion through social networks: an embedded cascade model[C]//Proceedings of WSDM. : ACM, 2016.

[124] Bourigault S, Lagnier C, Lamprier S, et al. Learning social network embeddings for predicting information diffusion[C]//Proceedings of WSDM. : ACM, 2014.

[125] Gao S, Pang H, Gallinari P, et al. A novel embedding method for information diffusion prediction in social network big data[J]. IEEE Transactions on Industrial Informatics, 2017.

[126] Li C, Ma J, Guo X, et al. Deepcas: An end-to-end predictor of information cascades[C]// Proceedings of WWW. 2017.

[127] Hu W, Singh K K, Xiao F, et al. Who will share my image?: Predicting the content diffusion path in online social networks[C]//Proceedings of WSDM. : ACM, 2018: 252–260.

[128] Wang J, Zheng V W, Liu Z, et al. Topological recurrent neural network for diffusion prediction [C]//Proceedings of ICDM. : IEEE, 2017: 475–484.

[129] Kefato Z T, Sheikh N, Montresor A. Di: Diffusion network inference through representation learning[M]. 2017.

[130] Cui P, Jin S, Yu L, et al. Cascading outbreak prediction in networks: a data-driven approach [C]//Proceedings of SIGKDD. : ACM, 2013.

[131] Tsur O, Rappoport A. What's in a hashtag?: content based prediction of the spread of ideas in microblogging communities[C]//Proceedings of WSDM. : ACM, 2012: 643–652.

[132] Weng L, Menczer F, Ahn Y Y. Predicting successful memes using network and community structure.[C]//ICWSM. 2014.

[133] Pinto H, Almeida J M, Gonçalves M A. Using early view patterns to predict the popularity of youtube videos[C]//Proceedings of WSDM. 2013.

[134] Gao S, Ma J, Chen Z. Modeling and predicting retweeting dynamics on microblogging platforms [C]//Proceedings of WSDM. : ACM, 2015.

[135] Cao Q, Shen H, Cen K, et al. Deephawkes: Bridging the gap between prediction and understanding of information cascades[C]//Proceedings of CIKM. : ACM, 2017.

[136] Goldenberg J, Libai B, Muller E. Talk of the network: A complex systems look at the underlying process of word-of-mouth[J]. Marketing letters, 2001.

[137] Saito K, Nakano R, Kimura M. Prediction of information diffusion probabilities for independent cascade model[C]//Knowledge-based intelligent information and engineering systems. : Springer, 2008: 67–75.

[138] Saito K, Kimura M, Ohara K, et al. Learning continuous-time information diffusion model for social behavioral data analysis[C]//Asian Conference on Machine Learning. : Springer, 2009: 322–337.

[139] Wang S, Hu X, Yu P S, et al. Mmrate: inferring multi-aspect diffusion networks with multi-pattern cascades[C]//Proceedings of SIGKDD. : ACM, 2014: 1246–1255.

[140] Celma Herrada Ò. Music recommendation and discovery in the long tail[M]. : Universitat Pompeu Fabra, 2009.

[141] Opsahl T, Panzarasa P. Clustering in weighted networks[J]. Social networks, 2009.

[142] Hodas N O, Lerman K. The simple rules of social contagion[J]. Scientific reports, 2014, 4: 4343.

[143] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[C]//Proceedings of ICLR. 2014.

[144] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of NIPS. 2017: 5998–6008.

[145] Rodriguez M G, Balduzzi D, Schölkopf B. Uncovering the temporal dynamics of diffusion networks[C]//Proceedings of ICML. 2011.

[146] LeCun Y, et al. Lenet-5, convolutional neural networks[J]. URL: http://yann. lecun. com/exdb/lenet, 2015.

[147] Van den Oord A, Dieleman S, Schrauwen B. Deep content-based music recommendation[C]// Proceedings of NIPS. 2013: 2643–2651.

[148] Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning[C]//Proceedings of ICML. : ACM, 2008.

[149] Kingma D, Ba J. Adam: A method for stochastic optimization[C]//Proceedings of ICLR. 2015.

[150] Ba J L, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016.

[151] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of CVPR. 2016: 770–778.

[152] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 15(1): 1929–1958.

[153] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735–1780.

[154] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations[C]// Proceedings of SIGKDD. 2014.

[155] Islam M R, Muthiah S, Adhikari B, et al. Deepdiffuse: Predicting the'who'and'when'in cascades [C]//Proceedings of ICDM. 2018.

[156] Wang Z, Chen C, Li W. A sequential neural information diffusion model with structure attention [C]//Proceedings of CIKM. 2018.

[157] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[C]// Proceedings of NeurIPS. 2017.

[158] Feng S, Cong G, Khan A, et al. Inf2vec: Latent representation model for social influence embedding[C]//Proceedings of ICDE. 2018.

[159] Yang C, Sun M, Liu H, et al. Neural diffusion model for microscopic cascade prediction[J]. arXiv preprint arXiv:1812.08933, 2018.

[160] Wang Y, Shen H, Liu S, et al. Cascade dynamics modeling with attention-based recurrent neural network.[C]//Proceedings of IJCAI. 2017.

[161] Wang Z, Chen C, Li W. Attention network for information diffusion prediction[C]//Proceedings of WWW. 2018.

[162] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[C]// Proceedings of ICLR. 2017.

[163] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine learning, 1992, 8(3-4): 229–256.

[164] Zhong E, Fan W, Wang J, et al. Comsoc: Adaptive transfer of user behaviors over composite social network[C]//Proceedings of SIGKDD. 2012.

[165] Yang C, Sun M, Yi X, et al. Stylistic chinese poetry generation via unsupervised style disentanglement[C]//Proceedings of EMNLP. 2018: 3960–3969.

[166] Mintz M, Bills S, Snow R, et al. Distant supervision for relation extraction without labeled data [C]//Proceedings of ACL. : Association for Computational Linguistics, 2009: 1003–1011.

致　谢

衷心感谢我的导师孙茂松教授对我的悉心指导，从您的言传身教中我受益匪浅。感谢刘知远老师在学术研究和工作规划上的巨大帮助。感谢赵德丽老师、赵鑫老师和唐建老师的学术指导和帮助。感谢每一位论文合作者的帮助。感谢我的父母在生活上给予的充分支持，让我能够心无旁骛的投身学术。感谢女朋友一直以来的陪伴，陪我度过所有的开心和不开心。感谢 THUNLP 的每一位小伙伴，五年的博士生活因大家的欢声笑语而不再枯燥。感谢每一位爱我的、支持我的人，谢谢你们！

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____ 日　期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

1992 年 1 月 31 日出生于天津市。

2010 年 9 月考入清华大学交叉信息研究院，2014 年 7 月本科毕业并获得工学学士学位。

2014 年 9 月免试进入清华大学计算机科学与技术系攻读博士学位至今。

发表的学术论文

[1] **Cheng Yang**, Jian Tang, Maosong Sun , Ganqu Cui , Zhiyuan Liu. Multi-scale Information Diffusion Prediction with Reinforced Recurrent Networks. International Joint Conference on Artificial Intelligence (IJCAI 2019). **(CCF A)**

[2] **Cheng Yang**, Maosong Sun, Zhiyuan Liu, Cunchao Tu. Fast Network Embedding Enhancement via High Order Proximity Approximation. International Joint Conference on Artificial Intelligence (IJCAI 2017). **(CCF A)**. Google Scholar 累计引用 **42** 次（2019 年 6 月）

[3] **Cheng Yang**, Maosong Sun, Wayne Xin Zhao, Zhiyuan Liu, Edward Chang. A Neural Network Approach to Joint Modeling Social Networks and Mobile Trajectories. ACM Transactions on Information Systems (ACM TOIS), 2017. **(CCF A)**. Google Scholar 累计引用 **28** 次（2019 年 6 月）

[4] **Cheng Yang**, Zhiyuan Liu, Deli Zhao, Maosong Sun, Edward Chang. Network Representation Learning with Rich Text Information. International Joint Conference on Artificial Intelligence (IJCAI 2015). **(CCF A)**. Google Scholar 累计引用 **303** 次（2019 年 6 月）

[5] **Cheng Yang**, Maosong Sun, Xiaoyuan Yi, Wenhao Li. Stylistic Chinese Poetry Generation via Unsupervised Style Disentanglement. Conference on Empirical Methods in Natural Language Processing (EMNLP 2018). **(CCF B)**

[6] **杨成**, 刘知远, 唐建. 信息网络表示学习的研究进展. 中国计算机学会通讯. 第 13 卷第 11 期.

[7] **Cheng Yang**, Maosong Sun, Haoran Liu, Shiyi Han, Zhiyuan Liu. Neural Diffusion Model for Microscopic Cascade Prediction. Submitted to IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE). **(CCF A)**

[8] Xiangkai Zeng, **Cheng Yang**, Cunchao Tu, Zhiyuan Liu, Maosong Sun. Chinese LIWC Lexicon Expansion via Hierarchical Classification of Word Embeddings with Sememe Attention. Association for the Advancement of Artificial Intelligence (AAAI 2018). **(CCF A)**

[9] Ayana, Shiqi Shen, Yun Chen, **Cheng Yang**, Zhiyuan Liu, Maosong Sun. Zero-shot Cross-Lingual Neural Headline Generation. IEEE Transactions on Audio, Speech and Language Processing (IEEE TASLP). **(CCF B)**

[10] Wei Chen*, Tian Lin*, Cheng Yang*. Real-time Topic-aware Influence Maximization Using Preprocessing. Computational Social Networks Vol. 3 (1), 8. (* 代表同等贡献)

[11] 涂存超 *, **杨成** *, 刘知远, 孙茂松. 网络表示学习综述. 中国科学: 信息科学, 2017, 47:980-996. (* 代表同等贡献)

专利

- 孙茂松, **杨成**, 矣晓沅, 李文浩. 一种具有风格多样性的绝句生成方法及装置. 专利号 201810420813.4. 实质审查阶段.
- **杨成**, 孙茂松, 刘知远, 涂存超. 一种网络节点的增强表示方法及装置. 专利号 201710354785.6. 实质审查阶段.