面向社会计算的网络表示学习

（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：涂　存　超
指 导 教 师：孙　茂　松　教　授

二〇一八年六月

Network Representation Learning for Social Computing

Dissertation Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Doctor of Philosophy

in

Computer Science and Technology

by

Tu Cunchao

Dissertation Supervisor : Professor Sun Maosong

June, 2018

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

在数据挖掘和社交网络分析中，对于网络节点的特征表示一直至关重要。随着大规模社会网络的出现，传统的网络表示方法面临着计算效率以及可解释性的问题。此外，这些社会网络往往蕴含着丰富的异构信息，这些特点使得已有的网络表示方法不能很好的处理这些大规模社会网络。

网络表示学习（Network Representation Learning），也就是网络嵌入（Network Embedding），目的是为网络中的节点学习一个低维实值的向量表示。每个节点对应的表示向量蕴含了该节点的网络结构信息以及其它异构信息，这些表示向量一般被当作特征向量，来进行进一步的网络分析任务，例如节点分类、链接预测、社区发现等。本文针对网络节点表示已有工作的不足，提出了在社会网络中学习节点显式及隐式表示的思路，来学习高质量的网络节点特征向量和提高社交网络分析任务的效果。为了学习网络节点显式的特征表示，我们进行了如下工作：（1）基于词项的显式网络表示：针对网络节点分类任务，我们提出一种双层分类模型，融合利用社交网络用户异构文本信息和网络结构信息，来进行职业预测任务。（2）基于主题标签的显式网络表示：为了提高用户特征表示的可解释性问题，我们提出利用显式的标签来表示用户节点，探究标签与社交网络用户社交行为之间的对应关系，进行用户标签推荐任务。

虽然网络节点显式表示可解释性强，但它面临着计算效率的问题。基于表示学习在图像、语音、文本等领域成功应用，我们提出了一系列基于深度学习的网络表示学习的方法，来学习网络节点的隐式低维表示。这些工作包括：（1）基于最大间隔的隐式网络表示：为了提高网络节点表示的区分性及其在节点分类上的效果，提出基于最大间隔理论的有区分性的网络表示学习模型，同时训练网络表示学习模型和最大间隔分类器，显著提升了网络节点分类的效果。（2）上下文相关的隐式网络表示：针对链接预测任务，提出上下文相关的网络表示学习模型，根据网络节点交互的邻居节点的不同，结合文本信息来学习节点动态的表示向量。由于引入了互相注意力机制，该模型能够显著提高链接预测任务的效果。（3）面向社会关系抽取的隐式网络表示：为了更好的对节点之间边上的语义信息进行建模，提出基于平移思想的网络表示学习模型，考虑节点之间边上的标签信息，来进行社会关系抽取任务。（4）社区优化的隐式网络表示：为了考虑社会网络中全局的社区特征，我们利用网络中的社区与文本中的主题之间的类比关系，提出了社区优化的网络表示学习模型，来同时学习节点表示和社区发现。

关键词：网络表示学习；网络嵌入；用户画像；标签推荐；节点分类；链接预测；社区发现

Abstract

How to represent vertices in networks plays important role in the fields of data mining and social network analysis. With the advent of large-scale social networks, typical network representation methods usually suffer from the issues of computational efficiency and interpretability. Besides, these social networks always contain abundant heterogeneous information. These characteristics make existing methods unsuitable to handle the large-scale social networks.

Network Representation Learning (NRL), i.e., Network Embedding (NE), aims to learn a real-valued low-dimensional representation vector for each vertex. These representation vectors contain the network structure and other heterogeneous information of vertices, and are usually treated as features in further network analysis tasks, including vertex classification, link prediction, community detection, and so on. To address the computational efficiency and interpretability issues of existing NRL methods, we propose to learn explicit and implicit network representations to improve the performance of network analysis tasks. To learn explicit network representations, we conducted the following works: (1) Lexical item-based explicit network representation. To improve the performance of vertex classification, we present a cascaded two-level classification framework with community refinement to incorporate the heterogeneous text information and network structure information of users. The proposed model achieves promising performance in profession identification. (2) Tag and topic-based explicit network representation. To address the interpretability issue, we employ the explicit tags to represent user vertices and exploit the correspondence between tags and social behaviors for user tag suggestion.

Although the explicit representation is interpretable, it suffers from the computation efficiency issue. Motivated by the success of representation learning in images, speech, and natural language, we propose a series of NRL works to learn implicit low-dimensional representations of vertices. These works include: (1) Max-margin implicit network representation. We propose Max-Margin DeepWalk (MMDW) to learn discriminative network representations and improve the performance of vertex classification, by training the max-margin classifier and NRL model jointly. (2) Context-aware implicit network representation. We propose Context-Aware Network Embedding (CANE) to learn the

dynamic embeddings of a vertex according to the neighbors it interacts with. By employing the mutual attention mechanism, CANE significantly improves the performance of link prediction. (3) Social relation extraction based implicit network representation. We propose a novel translation-based NRL model, TransNet, to model the relations between vertices. With the consideration of the semantic labels on edges, TransNet outperforms existing methods on social relation extraction task. (4) Community-enhanced implicit network representation. To integrate the global community pattern in social networks, we utilize the analogy between topics in text and communities in networks, and propose Community-enhanced NRL (CNRL) model to learn vertex representations and detect communities simultaneously.

Key words: Network Representation Learning; Network Embedding; User Profiling; Tag Suggestion; Vertex Classification; Link Prediction; Community Detection

目 录

主要符号对照表

NRL	网络表示学习 (Network Representation Learning)
NE	网络嵌入 (Network Embedding)
G	网络
V	网络中的节点集合
E	网络中的边集合
v	网络中的一个节点
e	网络中的一条边
TFIDF	Term-Frequency - Inverse Document Frequency
SVM	支持向量机（Support Vector Machine）
LDA	隐狄利克雷分配（Latent Dirichlet Allocation）
SC	谱聚类（Spectral Clustering）
UGC	用户生成内容（User Generated Content）
NC	标准传导性（Normalized Conductance）
NS	负采样（Negative Sampling）
SRE	社会关系抽取（Social Relation Extraction）

第 1 章　引言

1.1　研究背景

根据维基百科定义[①]，网络（network）用来表示离散的物体之间对称或者不对称的关联关系。在计算机科学中，网络通常可以表示成一个包含节点和边的图（graph）。网络结构的数据能够天然的用来表示不同物体之间的关系，各式各样的网络结构在我们日常生活中非常普遍。例如，在社交媒体平台中，人与人之间的关注、好友关系可以构成典型的社交网络；论文与论文之间的引用关系会构成学术引用网络；Web 页面之间的超链接关系也构成了互联网上的网页链接网络。

图 1.1　代表性的社交媒体平台。

随着互联网的发展，大规模的社交媒体平台不断涌现，如图 1.1所示，比较有代表性的社交媒体平台包括国内的新浪微博、微信、知乎，国外的 Facebook、Twitter、Instagram、Linkedin 等。这些社交媒体平台吸引了海量的用户。在这些平台中，用户与用户之间的关注、好友关系形成了典型的社交网络。与传统网络相比，这些大规模社交网络包括以下几个特点：

- 社交网络与传统网络相比，规模更大，而且更加稀疏。如图 1.2所示，据数据统计网站 Statista[②]统计，截止到 2018 年 1 月，全球最大的社交媒体平台 Facebook 的月活跃用户达到 21.67 亿，而中国最大的社交平台微信，月活跃

① https://en.wikipedia.org/wiki/Network_theory
② https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

用户也达到了 9.8 亿。这些社交网络包含海量用户节点的同时，也变得更加稀疏，大部分用户节点往往仅有有限的几十到几百个邻居节点。大规模和稀疏性的特点，为面向这些社交网络的网络分析、社会计算任务造成了巨大的挑战。

- 在大规模社交网络中，除了用户与用户之间的网络结构之外，还存在着丰富的用户行为信息。例如，用户在这些平台中发布或转发的文本、图片、视频等类型的内容信息，用户自身的介绍、标签等个人信息，用户对其它内容的点赞、分享信息等等。这些海量的异构信息能够反映出用户的兴趣爱好、个人属性等重要信息，对于面向社交媒体的应用服务具有重要的价值。
- 针对这些大规模社交媒体的应用场景非常丰富。例如，针对社交媒体用户，我们可以利用用户行为信息等对其进行用户画像，判断用户的性别、年龄、职业等属性信息，以及他们的兴趣爱好；基于用户画像结果，可以对用户进行个性化推荐，来推荐他们可能认识的好友或者感兴趣的新闻、产品等。

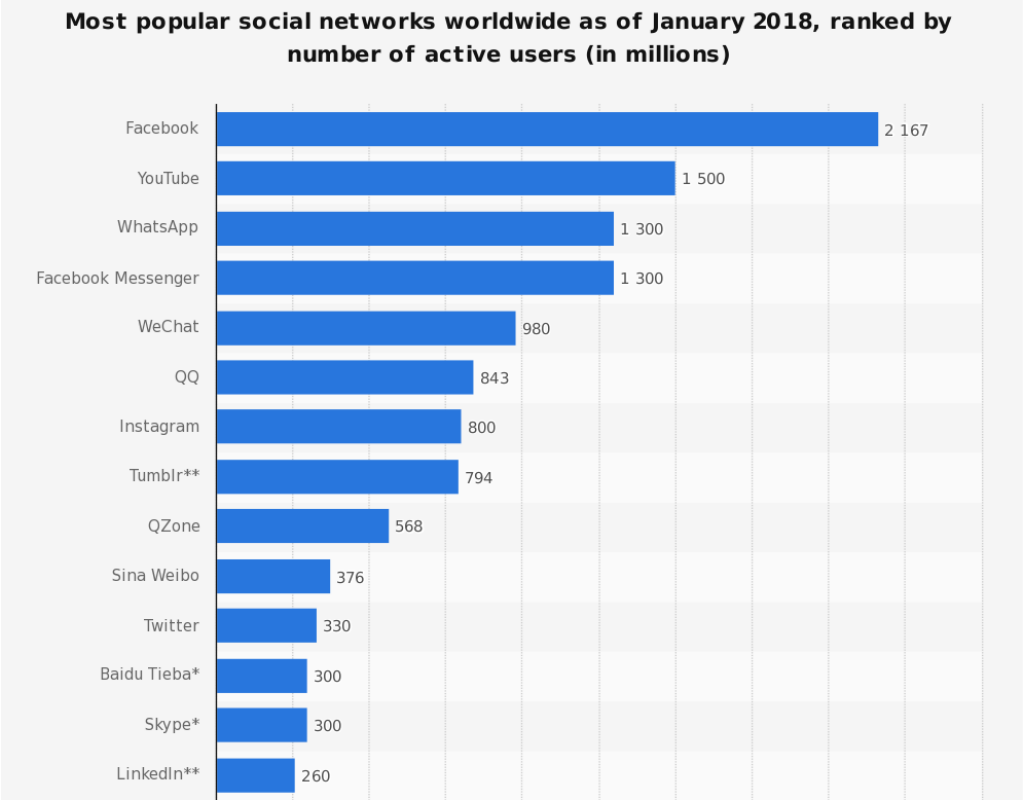


图 1.2　全球最大的社交媒体平台月活跃用户排名（截止到 2018 年 1 月）。

针对上述大规模社交网络的研究与应用近些年来成为计算社会科学、人工智能技术的热门研究领域。如何高效地在这些大规模社交网络上进行网络分析任务，例如节点分类、聚类、链接预测、社区发现等等，一直是该领域的研究基础和重

点。为了进行相应的网络分析任务，最重要的问题就是如何利用网络中的结构信息、异构信息，来对网络中的节点进行有效的特征表示，也就是如何进行网络表示。网络表示的质量，对于进行后续的网络分析任务至关重要。

1.2　基于符号的显式网络表示

最传统的对于网络节点的表示，叫做基于离散符号的表示，也就是指用一个高维稀疏的向量来表示网络节点。具体来说，对于一个包含有 N 个节点的网络，我们可以简单的用一个 $N \times N$ 的邻接矩阵来表示该网络，邻接矩阵的每一行为一个节点的邻接向量表示，对于相连的节点取值为 1 或者边上的权重，对于不相连的节点取值为 0。然而这种表示面临着维度过高以及稀疏性的问题。每一个表示向量的维度等于网络中的节点数量，而且该表示向量中大部分元素取值为 0。

同样的，对于网络节点附加的文本、标签等异构信息，我们同样可以依照词袋模型（bag-of-words），将这些信息中的离散元素（词、标签等）作为特征，构建类似的高维稀疏的表示向量，例如，tf-idf (term frequency–inverse document frequency) 向量。该表示向量的维度等于固定的词表的大小，而且向量中的大部分元素取值为 0。

这些高维的向量表示虽然具有良好的可解释性，对于该表示向量的每一个维度，都有直观含义，但是十分影响存储和计算效率，难以应用于大规模的社交网络计算中。

1.3　基于表示学习的隐式网络表示

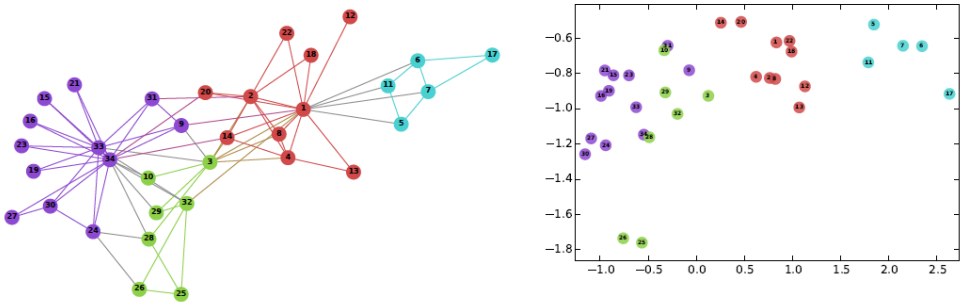


图 1.3　Karate 网络低维向量表示可视化[1]。

随着深度学习技术的发展，表示学习在图像[2]、语音[3]、自然语言处理[4] 等领域得到了广泛的研究和应用。表示学习是指通过学习的方式得到对象的分布式表

示，也就是低维实值的向量表示。随着表示学习的发展和成功运用，网络表示学习也逐渐成为数据挖掘领域的重要研究方向。如图 1.3所示，网络表示学习的目的是将网络中的节点映射到低维的表示空间，节点在表示空间的位置能够反映出该节点的网络结构信息，网络结构越相似的节点，在表示空间中的距离越接近。形式化的，给定一个网络 $G = (V, E)$，其中 V 表示网络中的节点集合，$E \subset V \times V$ 表示网络中节点与节点之间边的集合，网络表示学习的目的是为网络中的每个节点 $v \in V$ 学习一个低维实值的向量表示 $\mathbf{v} \in \mathbb{R}^k$，其中，$k$ 为表示空间的维度，而且 $k << |V|$。学习到的网络节点的表示向量，一般会作为特征向量，进行后续的社交网络分析任务，例如，节点相似度计算、节点分类、链接预测、社区发现等等。与基于符号的显式网络表示相比，基于表示学习的隐式表示通过用低维实值的向量来表示网络节点，虽然在可解释性上有所欠缺，但在网络分析任务中计算效率更高。通过利用先进的机器学习算法和深度学习技术对表示向量进行优化，使得这种隐式的表示向量往往质量更高，在后续的网络分析任务中性能更加优异。

下面我们将系统的介绍目前已有的几类网络表示学习方法。

1.3.1　基于矩阵特征向量的谱聚类方法

谱聚类 (Spectral Clustering) 方法是较早的用于学习网络节点表示的一类算法。具体来说，谱聚类方法会先根据网络节点之间的链接关系，构建关联矩阵，随后对该关联矩阵求特征值和特征向量，利用特征值最大的前 k 个特征向量，构建网络节点的 k 维表示。以下是谱聚类方法的代表性工作：

- 局部线性表示（Locally Linear Embedding, LLE）假设一个网络中任何一个节点的表示，都可以通过它的邻居节点的表示进行线性组合得到，这样使得降维之后的网络节点表示能够保留原有的网络拓扑结构，也就是保持流形的邻域不变的性质。LLE 把中心节点表示与邻居节点表示的线性组合之间的距离作为优化目标，最终将该问题的求解转化为对某个网络相关矩阵的特征值计算问题。
- 拉普拉斯特征映射（Laplacian Eigenmaps, LE）通过对网络对应的拉普拉斯矩阵进行特征值求解，来获得每个节点的低维向量表示。这里，拉普拉斯矩阵（Laplacian matrix）$L = D - A$，其中，D 为该网络的度矩阵，对角线之外的元素为 0，对角线上的值为节点的度数；A 为网络的邻接矩阵。LE 直接假设两个相邻的节点的表示向量应该尽可能接近。
- 有向图嵌入（Directed Graph Embedding, DGE）在拉普拉斯特征映射方法的基础上，利用随机游走算法（PageRank）对节点进行排序，来决定不同节点

的权重，从而赋予不同节点对应的损失函数不同的权重。

上述基于矩阵特征向量计算的网络表示方法，往往面临着三个问题：首先，这些方法的效果非常依赖于关联矩阵的定义和构建，不同的矩阵得到的网络表示在网络分析任务上的效果差别显著；此外，这些谱聚类的方法由于需要对关联矩阵进行特征值计算，时间复杂度较高；最后，对于大规模社交网络来说，其对应的关联矩阵大小为 N^2，需要占用极大的内存空间，空间复杂度也较高。上述三个问题使得谱聚类的方法往往不能适用于大规模社交网络的表示学习。

1.3.2　基于神经网络的方法

基于神经网络的表示学习方法由于其在自动特征学习方面的优势，在许多领域得到的成功的应用。在网络表示学习领域，近些年出现了许多工作，利用神经网络建模节点表示之间的关系，并利用随机梯度下降算法对网络节点表示进行优化学习。其中代表性的工作如下：

- DeepWalk[1] 是最经典的基于神经网络的网络表示学习方法。具体来说，DeepWalk 首先在网络上进行随机游走，来生成由节点构成的随机游走序列。随后，DeepWalk 通过将节点看作词，将节点序列看成句子，利用自然语言处理领域广泛使用的训练词向量的神经网络模型，Skip-Gram[5]，来训练网络节点的表示。DeepWalk 的随机游走算法和表示学习部分都能够通过并行算法进行加速，能够实现大规模社交网络的在线高效的表示学习。
- LINE[6] 是另外一个适用于大规模网络的网络表示学习模型。具体来说，LINE 定义了社交网络中节点之间的一阶邻近度（First-order proximity）和二阶邻近度（Second-order proximity）。对于直接相连的节点，LINE 利用两个节点表示之间的联合概率来刻画它们之间的一阶邻近度，也就是说相邻的节点表示应该尽量相似。而对于不直接相连的节点，LINE 假设如果它们的邻居越相似，那么它们的表示也应该越相近，因此引入节点的表示与邻居节点的上下文向量表示之间的条件概率来刻画节点之间的二阶邻近度。此外，LINE 算法能够有效的处理有向、无向以及加权的网络结构。
- node2vec[7] 模型是 DeepWalk 的扩展方法。通过改善 DeepWalk 模型的随机游走策略，node2vec 能够生成质量更高的节点序列。具体来说，node2vec 引入两个超参数 p 和 q，来控制随机游走算法的广度和深度。通过结合 BFS 和 DFS 搜索算法，node2vec 模型能够更好的对网络结构进行探索，使得网络节点表示既能够包含局部的网络结构信息，也能够包含更深层的全局的网络结构信息，从而节点表示质量更高，在节点分类任务上获得了显著的提升。

- SDNE[8]（Structural Deep Network Embedding）首次将典型深层神经网络引入网络表示学习中。与上述方法用到的浅层神经网络不同的是，SDNE 通过引入深层自动编码器（deep autoencoder）来对节点的邻接向量进行编码压缩，来得到节点低维实值的表示向量，并且保证节点的表示向量蕴含尽可能完整的邻居信息。

基于神经网络的方法一般需要设计合适的损失函数，通过随机梯度下降算法对模型中的参数进行优化。这一类方法与基于矩阵特征向量的方法相比，往往速度更快，效率更高，在后续的网络分析任务中表现更好。

1.3.3　基于矩阵分解的方法

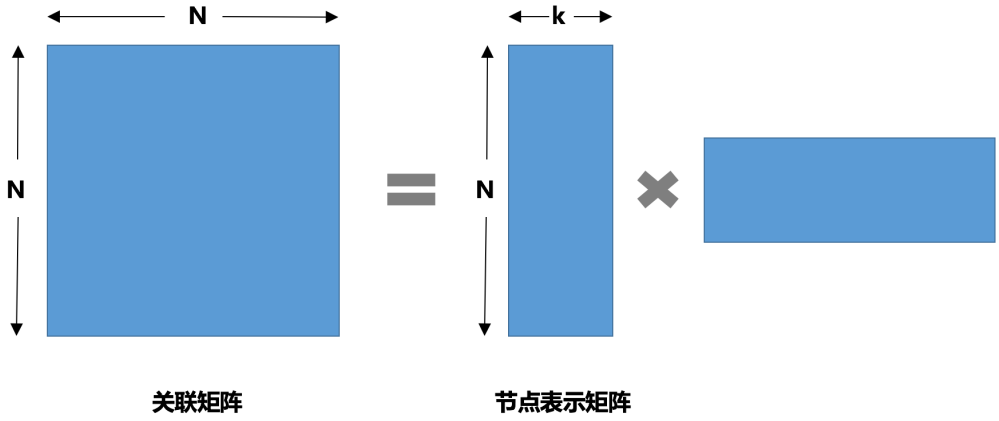


图 1.4　基于矩阵分解的网络表示。

基于矩阵分解的方法是另外一类重要的网络表示学习算法。如图 1.4所示，给定节点之间的关系矩阵，通过对该关系矩阵进行矩阵分解，来实现矩阵的降维，利用分解得到的低维矩阵，来作为网络节点的低维向量表示。具体来说，基于矩阵分解的网络表示学习有以下代表性方法：

- GraRep[9] 通过对邻接矩阵构成的关联矩阵进行矩阵分解，来得到考虑不同邻近度关系的节点表示。具体来说，GraRep 首先根据网络的邻接矩阵 A，来构建节点之间 t 步的关联矩阵 $M = A^t$，该矩阵中的每个元素 $M_{i,j}$ 表示由节点 v_i 经过 t 步的随机游走到达节点 v_j 的概率。通过对该关联矩阵 M 进行 SVD 分解，来得到包含节点 t 阶邻近度的向量表示。在此基础之上，都过使用不同的 t 值，将得到的不同 t 阶邻近度的向量表示进行拼接，可以组成更高维度的蕴含多阶邻近度信息的节点表示向量。但在 GraRep 中，需要通过矩阵相乘计算高阶的关系矩阵 A^t，对于大规模网络来说面临着计算效率和存储效率的问题。

- TADW[10] 通过引入文本矩阵，对关联矩阵进行 SVD 矩阵分解。TADW 首先证明了原始的 DeepWalk 模型等价于对关联矩阵 M 的矩阵分解。具体来说，关联矩阵 M 中的每个元素为：

$$M_{i,j} = \log \frac{[e_i(A + A^2 + \ldots + A^t)]_j}{t}. \tag{1-1}$$

  其中，t 表示 DeepWalk 采用的 Skip-Gram 模型中的窗口大小；e_i 为指示向量，该向量第 i 位为 1，其余位置为 0。实际应用中，对矩阵 M 的精确计算的复杂度为 $O(N^3)$。因此，一般情况下，采用近似计算的方法来得到该关联矩阵，也就是 $M = (A + A^2)/2$。在基于矩阵分解形式的 DeepWalk 模型基础上，TADW 通过引入文本矩阵，来对 M 矩阵进行 SVD 分解，得到包含节点文本信息的网络节点表示。
- 在 Qiu 等人[11] 的工作中，作者证明了几个经典的网络表示学习模型，都等价于对于特定关联矩阵的矩阵分解，这些方法包括 DeepWalk[1]、LINE[6]、PTE[12] 以及 node2vec[7]。

基于矩阵分解的网络表示学习工作一般关注于关联矩阵的构建，但面临着与谱聚类方法同样的问题，对于关联矩阵的计算和存储效率较低，难以适用于大规模的社交网络。

1.4　网络表示面临的挑战

在上述小节中，我们详细介绍了目前已有的网络表示方法及其各自的特点。总结来说，传统的网络表示方法面临着以下问题：

- **异构信息（heterogeneous information）**：对于真实世界的社会网络来说，网络中的节点往往拥有丰富多样的异构信息。例如，在社交网络中，用户的个人介绍信息，发布的微博，转发的消息、图片、视频等等；在学术网络中，研究者、论文等都拥有额外的文本、标签等信息。这些丰富的异构信息对于学习高质量的网络节点表示非常重要，然而没有被很好的融入到目前的网络表示的工作中。
- **可解释性（interpretability）**：一个好的网络节点表示除了应该在网络分析任务中表现优异之外，它的可解释性也非常重要。可解释性意味着对于人来说，能够直观的理解节点表示向量每一维所代表的含义，或者理解节点表示对于后续任务的影响所在。基于符号的显式网络表示一般具有较好的可解释性，而基于表示学习的隐式网络表示在这方面存在着明显的不足。

- **计算效率（computational efficiency）**：为了适用于大规模的社会网络分析任务，网络节点表示应该在模型训练以及实际使用上有着充分的效率保证。虽然基于符号的显式网络表示方式具有良好的可解释性，但是由于表示维度过高，在实际应用场景中，往往存储和计算效率较低；而基于表示学习的隐式表示方法，一般利用低维的实值向量来表示网络节点，存储和计算效率较高。同时，隐式网络表示所使用的模型以及梯度下降优化算法，往往只需要使用有限的存储空间，并且支持并行或者在线训练。

针对上述三个挑战，本文分别从网络节点的显式、隐式表示着手，探索如何有效的针对社会网络节点进行表示，并且针对具体的场景，融合网络中的多源异构信息，来提高网络节点表示在属性预测、社会标签推荐、节点分类、链接预测、社会关系抽取、社区检测等典型网络分析任务上的效果。

1.5　本文主要工作内容

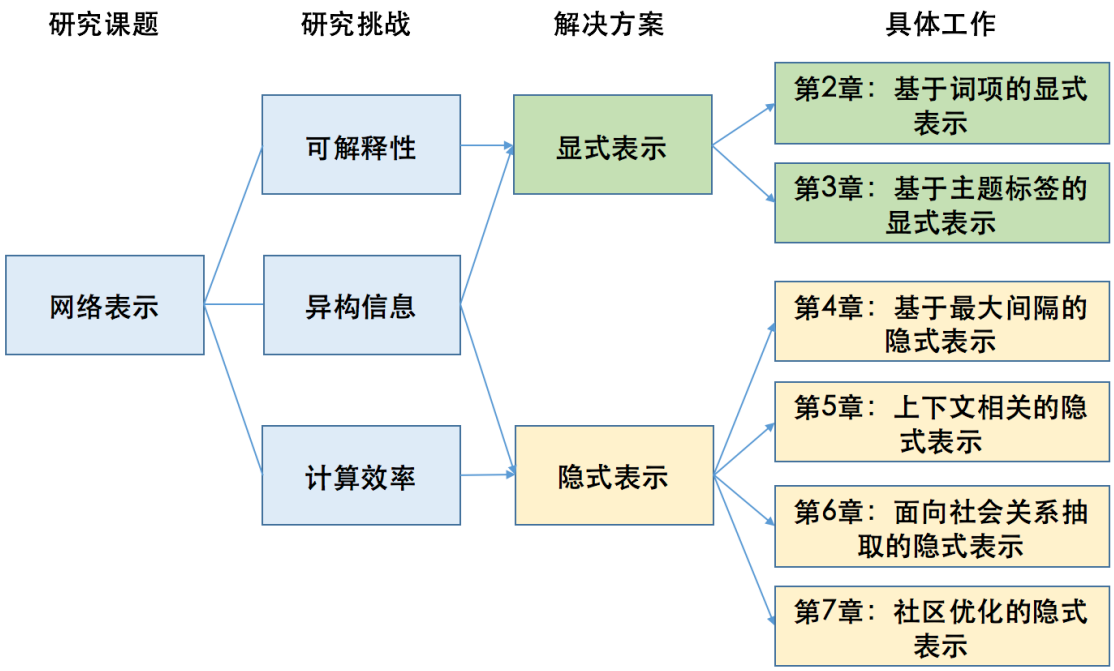


图 1.5　本文工作框架。

如图 1.5所示，本文针对网络表示面临的可解释性、异构信息以及计算效率的三个挑战，分别提出了显式表示和隐式表示两种解决方案。对于网络节点的显式表示，我们需要显式的定义网络节点表示每一维度的含义，从而使得这种表示方式具有很好的可解释性。为了进一步融合社交网络用户的多源异构信息，提升网络分析任务的效果，我们在显式网络表示方面进行了如下两个工作：

- **基于词项的显式网络表示**：针对社交网络用户丰富的异构信息，我们探究利用不同信息源中显式的词项，来对用户节点进行显式的表示。为了验证这种基于词项的用户表示的效果，我们选取用户属性预测中典型的职业预测任务，构建了融合用户个人异构信息的双层分类模型，并结合未标注数据和社区结构信息，来优化职业预测的效果。
- **基于标签主题的显式网络表示**：针对社交网络用户的多源异构信息，我们提出用显式的标签来对用户进行特征表示。这些标签能够反映出用户的个人兴趣或属性，具有较强的可解释性。此外，针对社交网络用户标签缺失的问题，我们对标签与用户异构信息之间的关联关系进行建模，能够有效的根据用户的这些信息为其推荐标签，得到未标注标签用户的基于标签的显式表示。在社会标签推荐任务上，我们提出的模型取得了显著的提升。

虽然显式表示方案具有很好的可解释性，但是往往面临着存储和计算效率的问题。此外，这种人工定义特征的方式往往可扩展性较差，不能很好的直接应用于其它场景。随着表示学习技术的发展，我们提出为网络中的节点学习低维实值的向量表示，也就是隐式的网络表示。这种表示方案存储和计算效率高，通过采用先进的深度学习模型和优化算法，所学习到的网络节点表示质量更高，在网络分析任务中效果更加出色。在网络的隐式表示方面，我们分别进行了如下四个工作：

- **基于最大间隔的隐式网络表示**：为了提高网络节点表示在最典型的网络分析任务，也就是节点分类上的效果，我们提出了一个半监督的基于最大间隔理论的网络表示学习模型，Max-Margin DeepWalk (MMDW)。该方法能够充分的利用网络节点的标签类别信息，来为其学习有区分性的网络节点表示。这种基于最大间隔的隐式表示方法，在节点分类任务上显著优于传统的网络表示学习模型。
- **上下文相关的隐式网络表示**：为了提高网络节点表示在链接预测任务上的效果，我们提出了一个上下文相关的网络表示学习模型，Context-Aware Network Embedding (CANE)，来利用网络节点附加的文本信息，学习节点的低维向量表示。具体来说，我们假设网络节点在与不同的邻居节点进行交互时，会展示不同方面的特点，因此需要与它的上下文邻居相关的表示。通过引入互相注意力机制，该方法能够有效的对节点之间的链接关系进行建模和解释。实验结果表明，CANE 在链接预测任务上取得了一致且显著的提升。
- **面向社会关系抽取的隐式网络表示**：为了对网络节点之间的关系进行显式的建模和预测，我们提出了面向社会关系抽取的网络表示学习模型，TransNet。该模型利用平移机制对节点和边的表示之间的关系进行建模，通过引入边上

的多标签语义信息，来获得节点的表示以及关系的表示。此外，该模型还能够对未标注标签的边进行关系预测，也就是社会关系抽取。在真实的大规模社会网络数据集上的实验结果表明，TransNet 与传统方法相比提升非常显著。

- **社区优化的隐式网络表示：**除了网络中附加的异构信息之外，网络结构本身也包含重要的全局特征，也就是社区信息。因此，我们提出了社区优化的网络表示学习模型，Community-enhanced NRL (CNRL)，来利用网络全局的社区信息优化网络节点表示。CNRL 利用网络中的社区与文本中的主题之间的类比关系，来同时进行社区检测和网络节点表示学习。通过引入全局的社区信息，CNRL 在节点分类、链接预测以及社区发现三个典型的网络分析任务上取得了一致的提升。

为了缓解隐式表示方案可解释性差的缺点，我们在上述工作中采用了向量可视化、注意力机制、标签等方式对节点、链接关系进行直观的解释，来改善隐式网络表示的可解释性问题。

最后，我们对本文的工作进行总结，并对面向社会计算的网络表示研究未来可能的研究方向进行展望。

第 2 章　基于词项的显式网络表示

社交网络中的用户往往拥有丰富的行为信息，例如用户标注的标签信息、发布的微博、用户之间的好友关系等等。这些丰富的异构信息，对于用户的特征表示，以及后续的用户属性预测任务非常重要。在本章的工作①中，我们通过社交网络用户异构信息中的词项，来对用户进行特征表示，并利用基于词项的用户表示进行属性预测任务，来验证这种用户表示方式的效果。

具体来说，我们选取社交媒体用户的职业信息作为用户的代表性属性，通过训练基于用户多源异构信息的双层文本分类器，来预测用户的职业属性。此外，我们还考虑了未标注数据的重要性，利用未标注数据不断扩充训练数据，迭代的优化职业分类器。最终，我们在分类器结果的基础上，考虑网络中的社区信息，来进一步优化职业分类器的预测结果。

2.1　问题描述

随着大规模社交网络的出现，用户可以方便的在微博等社交媒体平台彼此沟通交流。同时，这些社交媒体平台上的用户会互相关注，形成典型的用户社交网络。除了发布信息和彼此关注，社交媒体平台用户也会通过文本描述、标签等方式来介绍自己。这些用户生成内容（user-genenrated content, UGC）包含了用户丰富的信息，例如他们的社会属性和兴趣特征。利用社交媒体数据来研究用户的属性一直是社会学领域的研究热点。从性别、年龄等简单的用户属性[13]，到更复杂的性格[14]、幸福感[15]、政治倾向[16] 等。与上述用户属性相比，社交媒体中用户的职业属性一直没有被很好的研究过。

职业是人的非常重要的社会属性。社会学家一直在持续探究人的职业的特点。随着社交媒体的发展，职业也变成现代社会科学一个重要的研究课题。除了有利于社会学领域的研究，用户的职业信息也在实际场景中有着重要的应用，例如个性化推荐、广告投放等等。然而，大部分社交媒体平台用户的职业信息由于隐私关系一般不能够直接获取。因此，基于大规模社交网络数据来预测用户的职业属性，对于学术界和业界都有着重要价值。

用户的职业特征往往呈现在生活的方方面面。在社交媒体平台中，用户的职业信息也会显式或隐式的体现。因此，我们可以通过社交媒体平台的用户生成内

① 本章主要工作以 “PRISM: Profession Identification in Social Media’’ 为题发表在 2017 年的 “ACM Transactions on Intelligent Systems and Technology (TIST)’’ 上。

容来预测他们的职业信息。在本章的工作中，我们以新浪微博平台[①]为例，探究利用微博中的用户数据预测他们的职业信息。

在以微博为代表的社交媒体平台中，用户的职业信息往往体现在两个方面：

- 个人信息：微博用户往往提供个人简介、标签等信息来描述自己，此外，用户还会发布微博内容来表达观点。这些用户生成内容组成了用户的个人信息，为用户职业预测提供了重要的依据。
- 网络信息：一个用户往往会关注它感兴趣的其它用户。这种社交网络中的关注行为往往蕴含了与职业相关的社区信息。在真实的微博数据集中，我们将每个职业的用户看作一个社区，计算得到的模块性（modularity）[17] 高达 0.25。这也验证了职业信息与网络结构信息之间的较强的相关性，也验证社会科学中的同质性理论[18]，也就是越相似的用户越倾向于建立社交关系。

然而，进行社交媒体用户的职业预测仍然面临以下挑战：

- 用户生成的个人信息是高度异构的，如何将这些异构的个人信息进行融合？
- 与标注了职业信息的用户相比，存在数量更多的未标注用户。如何能够同时利用这两种用户数据，来帮助职业预测任务？
- 社交网络中相同职业的用户具有社区属性。如何有效的利用网络结构中的社区信息，来改进职业预测效果？

为了解决上述挑战，我们提出了一个有效的社交媒体用户职业识别的框架，**PR**ofession **I**dentification in **S**ocial **M**edia (PRISM)。PRISM 能够有效地利用用户的个人信息和社区结构信息，来识别他们的职业属性。

具体来说，首先，对于异构的个人信息，我们提出了一个双层的分类器，来衡量用户属于不同职业的置信度。在第一层分类器重，我们从不同的个人信息的信息源中抽取特征，也就是构建用户不同信息源的基于词项的表示，来训练不同信息源各自的基础分类器。随后，第二层分类器以第一层分类器各种的分类结果作为特征，来做出最终的预测结果。之后，我们根据 co-training 的思路，提出了多轮训练的迭代训练方法，来有效的利用标注数据和未标注数据，改善双层分类器的效果。最后，我们利用社区结构信息来调整最终的预测结果。

在实验部分，我们从新浪微博收集了 6 万多标注了职业信息的用户数据，作为我们的数据集。实验结果表明，我们的方法显著的优于其它基准方法，能够达到接近 85% 的职业预测准确率。利用训练好的模型，我们进一步探究了不同职业用户在用户信息、社交行为和语言风格等方面的特点和差异。

① http://weibo.com

2.2　相关工作

2.2.1　职业与社会科学

职业是一个非常重要的个人属性。作为社会过程中的重要因子，职业一直是社会学家关注的研究对象[19,20]。职业也与人的其它属性存在着关联关系，例如性格[21]。以往针对职业的研究工作，主要通过调查问卷的形式收集数据，因此，数据的规模十分受限。随着社交媒体的发展，用户在社交媒体中的行为数据为社会学研究提供了丰富信息。数据驱动的计算社会科学研究开始出现[22]，并且在许多题目上取得了成果，例如，性格[23]、幸福感[15] 和社会影响力[24] 等等。我们的职业预测的工作有助于与职业相关的社会学研究。

2.2.2　用户画像

用户画像的目标是针对社交媒体中的用户推测他们的不同属性[25]。这些属性可以大致被划分为两种类型，包括显式的属性（例如，性别、年龄、职业等等）和隐式的属性 (例如，兴趣、幸福感、政治倾向等等)。

已有的用户画像研究主要集中在显式的属性上，通常采用分类或者推荐的方法来进行属性预测。大部分基于分类的属性预测工作尝试从 UGC 中抽取有效的特征，来预测特定的属性，例如性别和年龄[13,26,27]、地理位置[16,28]、标签[29–34] 等等。大多数显式的属性可以通过用户生成的文本数据来进行预测。对于那些和社会性相关的属性，社交网络结构信息通常也会被考虑进来[28,35–37]。

研究者同样关注对于隐式属性的预测，例如，个人兴趣[36]、政治倾向[16]、性格[14,38] 以及社会权力[39] 等等。

本文工作主要集中在利用异构的文本信息和网络结构信息来预测社交网络用户的职业。对于社交网络用户的职业预测问题一直没有被很好的探究。与以往的属性预测工作相比，我们的框架能够有效的结合用户个人信息、网络社区信息以及未标注数据，共同帮助对于职业的预测。

2.3　模型框架

2.3.1　问题定义

假设社交媒体中的每个用户 $u \in U$ 可以表示成一个特征向量的集合 $\mathcal{X}_u = \{\mathbf{x}_{u,r}\}$。其中，$U$ 表示所有的用户集合，$\mathbf{x}_{u,r}$ 表示用户 u 的来自信息源 $r \in \{1, \ldots, R\}$ 的特征向量。这里，R 表示信息源的数量。除此之外，存在一个社交网络 $G = (U, E)$，其中，E 为用户之间的边的集合，也就是 $E \subset U \times U$。此外，我们的标注数据为

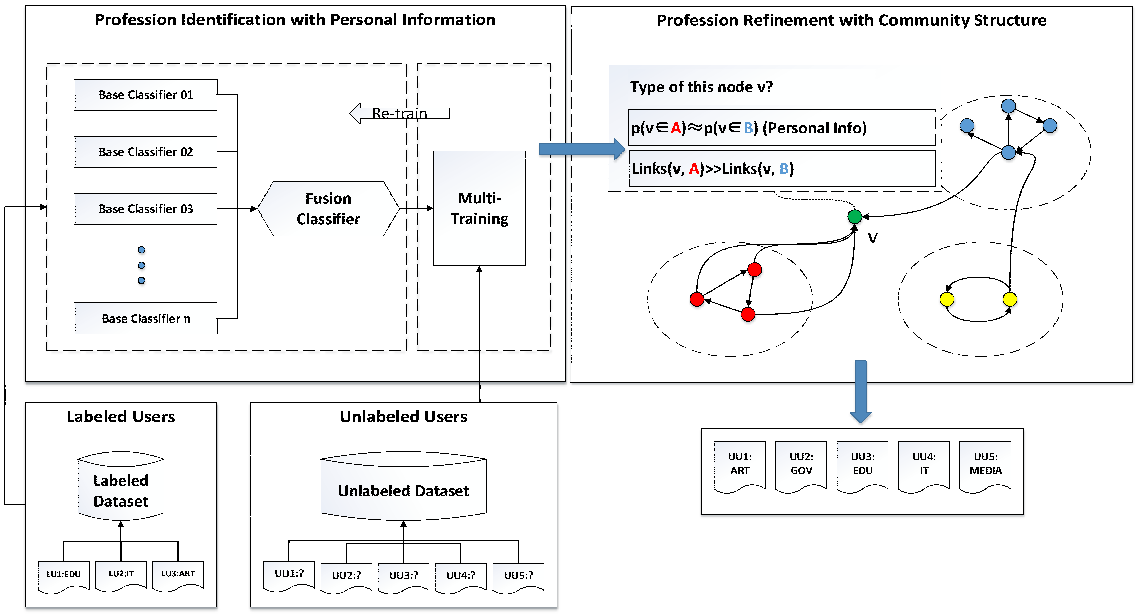


图 2.1　PRISM 模型框架。

$\{(\mathcal{X}_u, y_u)\}$，其中，$y_u \in \{1, \ldots, K\}$ 为用户对应的职业，K 是职业的数目。职业预测目的是对于未标注的用户，根据它们的个人信息和网络结构信息，预测它们对应的职业。

根据上述定义，我们设计了一个两步的职业预测过程：

(1) 我们将每个用户表示成多个从不同信息源抽取的特征向量。根据这些特征向量，我们构建一个双层分类器，来识别用户的职业。更进一步，我们利用多轮训练来结合未标注的数据，提高分类效果。

(2) 在第一步职业预测结果的基础上，我们利用职业社区的结构信息，来改善职业预测结果。

具体过程如图 2.1所示。在接下来的小节中，我们将详细介绍上述两个步骤的实现细节。

2.3.2　基于个人信息的职业预测

我们利用用户的个人信息，来构建双层分类模型，具体包括：

- **基础分类器构建：** 对于每个信息源 r，我们利用该信息源对应的特征向量和标签对 $\{(\mathbf{x}_{u,r}, y_u)\}$.，构建一个基础分类器 $f_r(\cdot)$。基于这些基础分类器，对于一个用户 u 和它不同信息源的特征向量集合 $\mathcal{X}_u$，我们能够得到一个识别矩阵 $\mathcal{P}_u = \{p_{k,r}\}$，其中 $p_{k,r} = \Pr(k|\mathbf{x}_{u,r}) = f_r(\mathbf{x}_{u,r}, k)$，表示第 r 个信息源对应的分类器认为用户 u 属于职业 k 的置信度。
- **基础分类器融合：** 我们将上述基础分类器输出的识别矩阵 $\mathcal{P}_u$ 作为新的特

征，来训练第二层的融合分类器 $g(\cdot)$。也就是说，融合分类器会根据基础分类器的预测结果，赋予不同分类器不同的权重，来得到最终的不同职业的置信度，也就是 $\Pr(k|\mathcal{P}_u) = g(\mathcal{P}_u, k)$。我们从中选取置信度最高的职业 $\hat{y}_u = \text{argmax}_k \Pr(k|\mathcal{P}_u)$ 来作为预测结果。

2.3.2.1　特征选取和基础分类器构建

在社交媒体平台中，用户通常有多种不同类型的信息。我们以新浪微博中的李开复为例。李开复是一个在中国有名的 IT 行业从业者。在新浪微博中，他提供了一个简短的个人简介 ‘‘创新工场 CEO’’，并且给自己标注了一些标签，例如 ‘‘风险投资’’、‘‘创新工场’’、‘‘教育’’、‘‘科技’’ 以及 ‘‘电子商务’’ 等。此外，他还拥有自己的认证信息 ‘‘创新工场董事长兼首席执行官’’。此外，他还发布了许多微博，这些微博包含了丰富的信息，例如，词、提及的用户、超链接、实体、hashtag 等等。这些不同的信息拥有各自的特点，而且都能反映出用户的个人属性信息。因此，在本章工作中，我们采用了 8 种不同类型的信息源，来作为基础分类器的特征。具体内容见表 2.1所示。

表 2.1　8 种不同的个人信息源。

No.	信息源	介绍
1	DES	用户的个人简介
2	TAG	用户的标签
3	VER	用户的认证信息
4	MSG	用户发布的微博
5	MEN	微博中提及的其它用户
6	URL	微博中提及的 url
7	ENT	微博中提及的命名实体
8	HAS	微博中提及的 hashtag

在这些特征源中，DES，VER 以及 MSG 都是文本形式，因此我们遵循词袋假设，将其中的词作为这些信息源对应的特征。其它的特征源都包含离散的元素，因此，我们直接把这些离散的元素看作该信息源对应的特征。

每个特征源都包含大量的候选特征。因此，我们需要进行特征筛选，来降低特征集合的规模。按照文本分类中进行特征筛选的做法[40,41]，我们采用卡方统计来选择每个特征源有代表性的特征。随后根据这些筛选后的特征构建基础分类器。

2.3.2.2　基础分类器融合

对于基础分类器输出的识别矩阵 $\mathcal{P}_u$，我们进行简单的拼接操作，将其转化成一个特征向量 $\mathbf{z}_u$，作为融合分类器输入的特征向量。这里，我们同样可以采用其它的方法将识别矩阵转化为一个特征向量，例如求平均或者求最大值。然而，实验结果表明，拼接操作的效果更好，因此我们在接下来的实验中，只汇报拼接的结果。

我们选用 Liblinear[42]①工具包来训练基础分类器和融合分类器。具体来说，我们选取 L2 正则的逻辑回归分类器 (L2R-LR)。

2.3.2.3　基于未标注数据的多轮训练

在社交网络中，未标注职业信息的用户规模要远大于标注用户的规模。因此，我们采用协同训练的思想，同时利用标注数据和未标注数据进行多轮训练。

具体来说，在训练完基础分类器之后，我们用这些基础分类器来识别未标注用户的职业，对于那些有一半以上的基础分类器分类结果一致的用户，我们将其作为新的标注数据加入训练数据中，然后重新训练基础分类器。

我们可以迭代的重复上述过程，直到基础分类器的效果不再发生变化。多轮训练可以丰富训练数据，因而能够提升分类的效果以及模型的泛化能力。

2.3.3　基于社区结构的结果优化

通过对数据的观察，我们发现，相同职业的用户之间更倾向于存在好友关系，以及形成社区，这与社会学理论中的同质性一致[18]。依照[35]，我们假设社交网络中相同职业的用户会形成一个跟职业相关的社区。在这种假设下，我们计算了社区的模块性指标 modularity[17]，这种社区划分对应的 modularity 为 0.25，验证了我们的猜想。

基于社区结构的职业预测结果优化过程如下。对于网络 $G = (U, E)$，我们根据标注用户，得到每个职业对应的子图，也就是社区 $G_k = (U_k, E_k)$，这里 U_k 表示属于第 k 个职业的用户集合，E_k 是该职业用户之间的边，也就是该职业对应的社区内部的边。随后，给定一个没有标注职业的用户集合 V，优化过程目的是将这些用户分配到正确的社区中，也就是预测其应该属于哪个职业对应的社区。这里的分配会受到加入该节点后，社区质量的变化的影响。

对于基于社区结构的结果优化来说，最重要的是如何定义一个对于每个职业社区的质量进行评测的指标。该指标应该能够反映出两方面的信息，一是网络结

① http://www.bwaldvogel.de/liblinear-java

构质量，也就是该社区内部的节点之间连接是否紧密，与外部节点连接是否稀疏；二是内容质量，也就是该社区中的根据个人信息预测得到的用户职业是否一致。

2.3.3.1　网络结构质量

为了衡量社区的网络结构质量，对于社区 $G_k = (U_k, E_k)$，我们定义 $U_{\neg k} = U \backslash U_k$，$E_{k,\neg k}$ 为 U_k 与 $U_{\neg k}$ 之间边的数量，$E_{k,k}$ 表示社区内部节点之间的边的数量，$E_{\neg k,\neg k}$ 为社区外部节点之间的边的数量。此外，我们定义 $E_k = E_{k,k} + E_{k,\neg k}$，$E_{\neg k} = E_{\neg k,\neg k} + E_{\neg k,k}$。

基于上述定义，社区 G_k 的结构质量可以形式化为：

$$Q_{structure}(G_k) = \frac{E_{k,k}}{E_{k,k} + E_{k,\neg k}} - \frac{E_k E_k}{E_k E_k + E_k E_{\neg k}}, \tag{2-1}$$

其中，第一项表示从社区内部节点出发的边，指向内部节点的比例；第二项表示一个随机图上述边的比例。这里，$Q_{structure}$ 取值范围为 $[-1, +1]$，取值越高，说明该社区 G_k 结构质量越高。

上述评测方式最早由 Mislove 等人[35] 提出，用来计算网络中社区的质量，叫作标准传导性，*normalized conductance*。

2.3.3.2　内容质量

内容的质量用来衡量社区内部节点在个人职业属性上的一致程度。我们定义一个社区的内容质量为所有属于该职业社区的用户在该职业上的置信度的平均，记为 $Q_{content}(G_k)$。利用内容质量，我们可以将基于个人信息的分类结果作为输入，来进行职业预测结果的优化。

2.3.3.3　职业预测优化

职业社区 G_k 对应的总体质量为 $Q_{structure}$ 和 $Q_{content}$ 的线性组合，如下所示：

$$Q(G_k) = \lambda Q_{structure}(G_k) + (1 - \lambda) Q_{content}(G_k), \tag{2-2}$$

其中，λ 为结构质量的权重系数。

利用上述 $Q(\cdot)$，我们采用了贪婪算法来进行社区扩展。给定一个职业 k，对于每个未标注的待划分的用户 $u \in V$，我们计算：

$$\Delta Q_k(u) = Q(G_k + u) - Q(G_k), \tag{2-3}$$

我们会找到一个 $\hat{u} = \arg\max_{u,k} \Delta Q(u)$，将该用户 $\hat{u}$ 加入到 U_k 中，重复该过程直到所有的未标注用户被划分。

在社区扩展之后，所有的未标注用户都被分配到了一个特定的职业社区中，我们将其对应的社区的职业来作为该用户的职业预测结果。

2.3.4　复杂度分析

在 PRISM 中，对于第 r 个特征源，训练分类器的复杂度为 $O(K|U|\cdot|\mathbf{x}_r|t_{LR})$，其中 t_{LR} 表示对于一个独立的特征和训练样例的平均计算时间。因此，双层分类器的复杂度为 $O(K|U|t_{LR}\sum_{r=1}^{R}|\mathbf{x}_r|)$。社区优化的复杂度为 $O(|V|^2K\cdot\frac{|E|}{|V|}) = O(K|V||E|)$。假设我们的多轮训练轮数为 m，那么 PRISM 模型总体的复杂度为 $O(mK|U|t_{LR}\sum_{r=1}^{R}|\mathbf{x}_r|+K|V||E|)$。这意味着，PRISM 模型的复杂度等于训练 mK 个 LR 分类器以及一个基于传导性的社区发现模型。

2.4　实验结果

2.4.1　数据集

我们从新浪微博收集了 62,415 个活跃的认证用户。这些用户被新浪微博官方，也就是新浪微博名人堂[①]，标注了 14 个不同的职业。此外，我们还利用 API 抓取了用户的个人信息以及超过一千万的微博。为了进行多轮训练，我们又收集了额外的 150,000 个未标注的认证用户。不同职业的用户所占的比例如表 2.2所示。

表 2.2　数据集中不同职业用户所占比例 (%)。

No.	职业	比例	No.	职业	比例
1	media（传媒）	25.6	8	education（教育）	4.0
2	government（政府官员）	15.1	9	fashion（时尚）	3.9
3	entertainment（娱乐）	8.8	10	games（游戏动漫）	3.8
4	estate（房地产）	8.2	11	literature（文学出版）	3.4
5	finance（财经）	7.0	12	services（服务）	3.4
6	IT	6.4	13	art（人文艺术）	3.1
7	sports（体育）	5.6	14	healthcare（健康医疗）	1.7

① http://verified.weibo.com/.

2.4.2　实验设置

我们将 62,415 个标注用户随机选取 4/5 作为训练集，剩下的作为测试集。我们选取 accuracy、macro-averaging precision/recall/F1 作为评测指标。

如前面所述，社交媒体用户职业预测没有被以往的工作研究过，因此，没有能够处理异构的特征源，直接可以用来进行职业预测的基准方法。为了进行对比，我们采用了只考虑单个信息源的职业预测模型以及进行特征拼接的方法作为基准方法。

对于特征筛选，我们评测不同特征数量下的基础分类器的效果，最后针对不同特征源选取表现最好的特征数量，分别为：DES2,300，TAG3,800，VER4,000，MSG6,600，MEN3,200，URL2,700，ENT3,600 以及 HAS4,100。我们公式 (2-2) 中的参数 λ 设为 $\lambda = 0.2$，此时模型有着最好的效果。

表 2.3　职业预测实验结果 (%)。

Method	Accuracy	Precision	Recall	F
DES	31.25	51.82	28.90	37.11
TAG	38.11	50.55	31.04	38.46
VER	78.63	75.73	74.89	75.31
MSG	47.47	49.58	42.79	45.93
MEN	38.22	42.85	30.59	35.70
URL	26.38	36.47	13.68	19.90
ENT	33.86	36.88	26.95	31.15
HAS	30.91	37.44	17.60	23.94
Single Vector	39.25	48.33	34.92	40.54
Fusion	81.25	79.60	76.27	77.90
Fusion+MT	**83.38**	**82.24**	**81.35**	**81.79**

2.4.3　实验结果和分析

表 2.3展示了使用不同特征源的职业预测结果。在该表中，“Single Vector’’ 表示将不同的特征源对应的特征向量直接拼接，来训练职业预测分类器的方法。“Fusion” 表示我们提出的双层分类器的方法，“Fusion + MT’’ 表示在双层分类器结果的基础上进行多轮训练的效果。从表 2.3中，我们发现：

- 我们提出的双层融合分类器效果要显著优于简单拼接特征向量的 “Single Vector’’ 方法。这表明双层分类器模型能够有效的融合异构的特征源。
- 基础分类器中，利用认证信息进行职业预测的效果要优于其它信息源的效果。这与事实一致，因为认证信息经过了人工审核，更有信息量，与其它特征源

相比，噪音更少。

- 双层的融合分类器的效果明显优于所有考虑单独信息源的基础分类器。这表明对单层分类器的结果进行融合的合理性，它能够有效的融合不同异构信息，提高职业预测的效果。
- 经过多轮训练之后，模型的职业预测结果在所有评测指标上均超过了 80%。这个结果一方面验证了利用多轮训练考虑未标注数据的有效性，也表明我们的模型对于不同类别的职业预测效果十分均衡。

2.4.4　基于职业社区的优化结果

表 2.4　考虑网络结构的职业预测结果 (%)。

Method	Accuracy	Precision	Recall	F
LPA	58.86	57.05	54.53	55.76
CD	64.20	65.11	60.78	62.87
PRISM				
$\lambda = 0.1$	84.17	83.15	81.62	82.37
$\lambda = 0.2$	**84.92**	**83.78**	**81.89**	**82.82**
$\lambda = 0.3$	81.12	79.10	77.42	78.25
$\lambda = 0.5$	77.56	76.53	75.08	75.79

为了衡量引入职业社区结构信息之后 PRISM 模型的效果，我们与两种典型的基于网络结构的节点分类模型进行对比，包括 label propagation algorithm (LPA)[43] 以及社区检测模型 CD[35]。这两种方法仅仅考虑网络结构信息进行用户节点的职业预测。

实验结果如图2.4所示，我们发现：

- 与前面的 Fusion+MT 方法相比，进一步的考虑社区结构信息使得模型获得了显著的提升。这表明网络结构信息也能够对职业预测任务提供重要的支持信息。根据我们的统计显示，在社交网络中，个人信息的缺失现象非常明显，例如，测试用户中 153 名用户没有认证信息，从而导致预测效果的下降。考虑额外的网络结构信息，能够处理这种个人信息缺失的问题。
- PRISM 模型显著的优于其它只考虑网络结构信息的模型。这表明了同时考虑用户个人信息及网络结构信息的必要性。此外，我们发现，当 $\lambda = 0.2$ 时，我们的模型取得了最好的效果。在实际应用中，我们可以进行交叉验证，或者设置验证集，来选取合理的参数取值。

2.4.5　错误分析

表 2.5　不同职业的预测结果 (%)。

No.	职业	Precision	Recall	F
1	media	84.04	90.60	87.20
2	government	94.03	93.78	93.90
3	entertainment	84.78	82.25	83.49
4	estate	88.22	86.92	87.57
5	**finance**	68.86	73.05	70.90
6	**IT**	72.93	68.38	70.58
7	sports	94.05	92.84	93.44
8	**education**	76.88	73.80	75.31
9	fashion	84.84	78.94	81.78
10	game	85.47	84.19	84.82
11	literature	84.68	75.99	80.10
12	**service**	65.32	57.45	61.13
13	**art**	76.84	69.92	73.22
14	healthcare	87.10	87.50	87.30

在表2.5中，我们展示了模型对于不同职业各自的预测结果。从该表中，我们发现，大多数职业都有着较高的预测效果，然而对于 “services”, “IT”, “finance”, “art’’ 以及 “education” 职业，模型效果相对较差。

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	91.05	0.66	1.88	0.81	1.05	1.43	0.51	0.30	0.24	0.30	0.39	0.95	0.42	0.03
2	2.64	93.45	0.17	0.39	0.84	0.28	0.03	0.78	0.02	0.00	0.05	0.17	0.39	0.78
3	9.72	0.47	81.34	0.19	1.43	0.66	0.28	0.19	2.09	0.57	1.15	0.38	1.53	0.00
4	5.31	0.40	0.26	82.76	5.82	1.04	0.40	0.26	0.51	0.00	0.26	2.71	0.26	0.00
5	3.95	2.22	0.29	3.56	77.37	6.55	0.29	1.92	0.77	0.29	0.29	2.03	0.39	0.09
6	7.78	0.44	0.00	2.52	9.42	72.18	0.33	0.88	0.55	4.06	0.11	1.43	0.11	0.22
7	3.45	0.36	0.36	0.36	0.79	0.00	94.05	0.24	0.00	0.12	0.00	4.00	0.24	0.00
8	3.89	3.05	0.85	0.68	4.40	1.19	0.34	77.82	0.34	0.00	0.85	3.72	1.19	1.69
9	5.93	0.18	4.14	0.54	2.88	0.90	0.36	0.72	81.31	1.26	0.18	0.90	0.72	0.00
10	3.19	0.00	1.60	0.18	0.00	5.14	0.71	0.00	1.60	86.70	0.35	0.18	0.35	0.00
11	9.33	1.87	1.65	0.00	1.45	0.00	0.00	2.28	0.20	0.83	78.65	0.00	3.32	0.42
12	13.23	1.04	1.86	4.34	6.41	4.76	0.00	3.72	1.04	0.83	0.00	62.56	0.21	0.00
13	4.71	2.35	4.71	0.00	2.35	0.00	0.79	2.95	0.79	0.79	3.33	0.20	77.05	0.00
14	2.64	0.76	0.37	0.00	2.64	0.37	0.00	1.13	0.75	0.00	1.51	0.00	0.01	89.82

图 2.2　不同职业的预测结果分布情况。

为了探究这些职业预测效果不理想的原因，我们在图 2.2中展示了不同职业预测结果的分布情况。在该图中，我们定义第 i 行第 j 列的结果表示属于职业 i 的用

户被划分为职业 j 的比例，也就是：

$$e_{ij} = \frac{\sum_{u \in U_i} k_u = j}{|U_i|}, \tag{2-4}$$

其中，k_u 表示对于用户 u 的职业预测结果。为了更直观的展示上述分布，我们根据不同的分布比例，给每个单元格设置了不同深度的背景颜色，绘制了该混淆矩阵的热度图。从该图中，我们发现：

- “service’’ 容易被错误划分到 “media’’ 这个职业，“art’’ 和 “education’’ 两种职业也容易被互相错误划分。这是由于，这些职业之间互相有重叠，职业之间的界限并不明显，因此造成了难以区分的问题。例如，“service’’ 职业的用户经常与 “media’’ 职业的用户进行交互，因为他们都对广告和市场非常关注。
- “finance’’ 和 “IT’’ 两个职业也容易发生互相混淆的情况。通过样例分析，我们发现，许多公司的高管同时具有这两个领域的经历，这种情况下对于他们的职业划分变得非常困难，这种混淆也会体现在他们的好友网络中，两个职业的用户之间存在着紧密的链接关系。

2.4.6　职业分析

为了探究微博用户的职业特点，我们利用训练好的 PRISM 模型，对大量未标注的用户进行职业预测，然后在这些预测结果的基础上，分析不同职业用户的特点。

2.4.6.1　用户统计

我们在表 2.6中展示了不同职业用户的统计信息。“Gender’’ 这列表示用户的性别比 ($\frac{\#\text{male}}{\#\text{female}}$)，“Message’’ 一列表示该职业用户平均发布的微博数量，“Follower’’ 这列表示用户平均的粉丝数。

从表 2.6中，我们发现一些直观的现象：

- 由于不同职业存在着各自的特点，不同职业之间存在着明显的性别差异。“IT”, “art”, “government”, “finance’’ 以及 “sports’’ 等职业的男女比高达 2.50 以上，然而只有 “fashion’’ 一个职业男女比低于 0.80。这个现象也验证了一些研究中的性别区分理论[44]，也就是说女性从事的职业范围更窄。
- 不同职业的用户平均发布的微博数量非常接近，然而一些公众职业的粉丝数要远大于其它职业的粉丝数，例如 “entertainment”, “sports”, “literature’’ 以及 “fashion’’ 等，这也符合我们对这些职业的直观认知。

表 2.6　不同职业用户数据统计。

No.	职业	性别	微博数量	粉丝数
1	media	1.25	1,776	18,555
2	government	2.66	1,270	17,724
3	entertainment	1.57	1,367	59,709
4	estate	2.42	1,375	6,661
5	finance	2.60	1,457	15,990
6	IT	3.19	1,558	17,224
7	sports	2.51	1,380	36,813
8	education	1.88	1,384	11,216
9	fashion	0.77	1,392	24,867
10	games	1.75	1,358	20,354
11	literature	0.94	1,924	31,522
12	services	2.04	1,512	7,964
13	art	2.68	1,667	14,447
14	healthcare	1.38	1,386	14,726

2.4.6.2　不同职业的社会网络

正如我们前面提到的，社交媒体中用户的职业与网络结构信息高度相关。这里，我们定义*职业吸引力*来衡量一个职业的用户与另外一个职业的用户成为好友的倾向，即 $a(i,j) = g\big(100 \times (\frac{N_{i,j}}{N_i} - \frac{U_j}{U})\big)$。这里，$g()$ 为 sigmoid 函数，N_i 为职业 i 的用户平均的好友数，在这些好友中，$N_{i,j}$ 表示属于职业 j 的好友的平均数量，$\frac{U_j}{U}$ 为全局的好友中职业为 j 的比例。$a(i,j)$ 的值越高，表示职业 i 的用户更有可能与职业 j 的用户成为好友。在图2.3中，我们展示了不同职业之间，在关注网络和好友网络（互相关注）上，职业吸引力的结果。

从该图中，我们发现：

- 不同职业的用户更倾向于与相同职业的用户存在好友关系，但是也会有一些明显的对于其它职业的偏好。
- 一些职业的用户，例如 “entertainment”, “education’’ 以及 “literature’’ 倾向于被其它用户关注，而 “media”, “government’’ 以及 “estate’’ 的用户倾向于关注其他职业的用户。
- “IT’’ 与 “finance/game/service’’ 之间，“entertainment’’ 与 “fashion/art’’ 之间存在着较强的相关性，表明了这些职业之间在社会领域的协作关系。
- 职业之间的吸引力有时是不对称的。例如，职业为 “eduction’’ 的用户倾向于关注 “art’’ 职业的用户，而反过来的关注倾向相对较弱。

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1.00	0.00	1.00	0.00	0.15	0.09	0.04	0.26	0.43	0.05	0.93	0.14	0.60	0.30
2	0.00	1.00	0.99	0.00	0.18	0.04	0.12	0.44	0.15	0.06	0.85	0.09	0.61	0.39
3	0.00	0.00	1.00	0.00	0.01	0.01	0.02	0.08	0.89	0.06	0.65	0.09	0.82	0.21
4	0.00	0.00	0.50	1.00	0.91	0.05	0.02	0.16	0.12	0.04	0.58	0.19	0.38	0.23
5	0.00	0.00	0.73	0.10	1.00	0.98	0.03	0.51	0.24	0.06	0.77	0.21	0.56	0.33
6	0.00	0.00	0.21	0.01	1.00	1.00	0.01	0.33	0.10	0.28	0.68	0.26	0.35	0.28
7	0.00	0.00	0.99	0.00	0.01	0.01	1.00	0.06	0.25	0.04	0.19	0.06	0.29	0.20
8	0.01	0.00	0.94	0.01	0.97	0.54	0.03	1.00	0.17	0.06	0.98	0.24	0.85	0.44
9	0.00	0.00	1.00	0.00	0.04	0.02	0.02	0.07	1.00	0.07	0.41	0.16	0.94	0.25
10	0.00	0.00	0.98	0.00	0.10	1.00	0.02	0.07	0.47	1.00	0.76	0.10	0.44	0.19
11	0.00	0.00	0.98	0.00	0.04	0.10	0.01	0.41	0.10	0.27	1.00	0.10	0.94	0.31
12	0.01	0.00	0.99	0.27	0.94	0.96	0.02	0.79	0.51	0.10	0.85	1.00	0.69	0.34
13	0.00	0.00	1.00	0.00	0.07	0.03	0.02	0.27	0.92	0.10	0.98	0.14	1.00	0.24
14	0.00	0.00	0.91	0.00	0.32	0.07	0.04	0.66	0.19	0.05	0.90	0.09	0.55	1.00

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	1.00	0.00	0.07	0.00	0.04	0.07	0.02	0.14	0.25	0.06	0.47	0.25	0.30	0.24
2	0.00	1.00	0.00	0.00	0.02	0.02	0.01	0.30	0.04	0.05	0.13	0.19	0.18	0.31
3	0.01	0.00	1.00	0.00	0.01	0.01	0.01	0.08	0.91	0.07	0.32	0.13	0.82	0.20
4	0.00	0.00	0.00	1.00	0.08	0.03	0.01	0.05	0.05	0.04	0.06	0.44	0.09	0.19
5	0.00	0.00	0.01	0.08	1.00	0.99	0.02	0.44	0.24	0.09	0.18	0.59	0.28	0.34
6	0.00	0.00	0.00	0.00	0.98	1.00	0.01	0.21	0.06	0.62	0.17	0.60	0.14	0.24
7	0.00	0.00	0.00	0.00	0.01	0.01	1.00	0.04	0.08	0.04	0.05	0.07	0.10	0.18
8	0.01	0.00	0.01	0.00	0.76	0.54	0.01	1.00	0.10	0.10	0.61	0.67	0.71	0.46
9	0.01	0.00	1.00	0.00	0.03	0.02	0.02	0.08	1.00	0.10	0.11	0.28	0.93	0.24
10	0.00	0.00	0.00	0.00	0.01	0.99	0.01	0.04	0.15	1.00	0.29	0.16	0.19	0.15
11	0.01	0.00	0.09	0.00	0.02	0.10	0.01	0.35	0.07	0.22	1.00	0.13	0.93	0.28
12	0.04	0.00	0.01	0.65	0.59	0.98	0.01	0.77	0.27	0.20	0.18	1.00	0.32	0.31
13	0.01	0.00	0.97	0.00	0.04	0.02	0.01	0.29	0.95	0.12	0.92	0.20	1.00	0.23
14	0.00	0.00	0.00	0.00	0.11	0.07	0.01	0.56	0.13	0.05	0.29	0.15	0.18	1.00

图 2.3　不同职业之间职业吸引力结果（上图为关注网络，下图为好友网络）。

2.4.6.3　不同职业的语言风格

语言学家一直认为，用户的用语习惯、语言风格能够反映出用户的属性特点[45,46]。同样的，不同职业的用户拥有各自行业的专业术语和语言风格。这里，我们对不同职业的用户对于虚词的使用习惯进行统计，来探究职业与用户语言风格之间的关系。

我们首先对于不同职业用户发布的微博进行词性标注，然后统计出不同职业的用户使用连词、感叹词和语气词的情况。如表 2.7所示，对于每一种虚词，我们用 $\triangle$ 标注出使用最频繁的几个职业，用 $\triangledown$ 标注出使用最少的几个职业。

我们可以发现，“healthcare”, “education”, “finance’’ 以及 “IT’’ 等行业的用户会使用更多的连词，使用更少的语气词和感叹词。这是因为这些职业的用户在工作中更加理性，更有逻辑性，用语更加准确。形成对比的是，“sports”, “entertainment”, “art’’ 以及 “media’’ 的用户会使用较少的连词，而使用更多的感叹词和语气词。这是因为，这四种职业需要人们更感性，更有想象力，不拘一格。这种语言风格的差异非常显著，也印证了社会语言学中不同职业用户的特点能够通过语言风格来反映的结论。

表 2.7　不同职业的用户使用虚词的比例统计 (%)。

No.	职业	连词	感叹词	语气词
1	media	1.19▽	0.22△	2.16△
2	government	1.29	0.17	1.70
3	entertainment	1.08▽	0.26△	2.38△
4	estate	1.26	0.15	1.72
5	finance	1.39△	0.15▽	1.65▽
6	IT	1.35△	0.15▽	1.66
7	sports	1.04▽	0.25△	2.60△
8	education	1.42△	0.16▽	1.55▽
9	fashion	1.25	0.22	1.95
10	games	1.34	0.16	1.26▽
11	literature	1.31	0.27△	2.25
12	services	1.29	0.18	1.94
13	art	1.11▽	0.22△	2.06△
14	healthcare	1.76△	0.11▽	1.15▽

2.5　本章小结

在本章工作中，我们考虑社交媒体用户多源异构的行为信息，利用基于词项的表示构建不同特征源的特征向量，来进行职业预测任务。我们提出的 PRISM 框架能够共同利用用户个人信息以及网络结构信息，来改进职业预测的效果。在真实数据集的实验结果表明，我们的方法能够有效的对于不同职业的用户进行职业预测。此外，为了分析不同职业用户的特点，我们还进行了不同职业用户信息统计、社会网络、语言风格差异的分析。

第 3 章　基于主题标签的显式网络表示

在前一章节中，我们介绍了用社交媒体用户节点的不同特征源的词项，来表示社交网络中的用户节点。虽然我们能够对于不同信息源的特征，通过双层分类模型进行融合，来帮助用户的属性预测，但是这种基于词项的用户节点表示多源异构，由于包含了多种来源的不同特征，不能对用户节点进行统一的表示。

我们发现，社交媒体中的用户可以给自己标注不同类型的标签，这些标签能够很好的反映出用户的个人兴趣或者属性。因此，在本章[①]中，我们提出用这些显式的标签来对社交网络用户节点进行显式的表示。同时，虽然社交媒体用户有着大量的行为信息，但是仍然存在标签缺失问题，因此，我们提出了标签关联模型（Tag Correspondence Model, TCM）来探究用户的标签与多源异构信息之间的关联关系。此外，TCM 模型能够根据这些多源异构的信息来帮助没有标签的用户进行推荐标签。实验结果表明，我们提出的标签关联模型 TCM 能够有效的识别出标签的关联关系，在用户标签推荐任务上取得了比基准方法更好的效果。

3.1　问题描述

微博是 Web2.0 时代的重要的信息平台。在微博中，用户可以发布或阅读短文本、图片、视频等信息，这些信息使得微博成为一个重要的用户发布信息和分享观点的平台。

在该平台中，用户有着丰富的行为信息，例如发布微博和进行评论。此外，微博用户之间的关注行为构成了一个复杂的用户社交网络。这些用户生成内容和社交网络信息构成了微博用户的上下文信息。

对于微博来说，如何根据用户的兴趣来推荐相关信息服务非常重要。为了更好的理解用户的兴趣，微博允许每个用户给自己标注一些标签。我们以新浪微博上的李开复为例，李开复作为 ‘‘创新工场’’（一个关注互联网领域的投资公司）的 CEO，他给自己标注了 ‘‘风险投资”, ‘‘微博粉丝”, ‘‘创新工场”, ‘‘教育”, ‘‘科技”, ‘‘电子商务”, ‘‘移动互联网”, ‘‘创业”, ‘‘互联网’’ 以及 ‘‘《世界因你不同》’’ 的标签。这些标签让我们有效以及直接的理解他的个人兴趣或者属性。

然而，因为用户的标签都是自己标注的，所以存在很多噪音和不规整的现象。为了更好的理解用户标注的标签的语义含义，我们尝试从微博用户丰富的上下文

① 本章主要工作以 “Tag Correspondence Model for User Tag Suggestion’’ 为题发表在 2015 年的 “Journal of Computer Science and Technology (JCST)’’ 上。

信息中找到标签的语义关联（Correspondence）。这里的 **correspondence** 是指上下文信息中与标签语义相关的一个单独的元素。例如，对于标签 ‘‘移动互联网’’ 来说，我们可以从李开复的微博中，找到相关的词 “IT’’ 作为这个标签的关联元素。

一般而言，微博用户的上下文信息有多个来源。每个来源都有各自的关联元素集合。这些来源可以被大致分为以下两种类型：

用户相关信息源：由用户自己生成的信息被称作用户相关特征源，例如用户发布的微博、个人信息等。这些用户生成内容能够反映出一个用户的兴趣爱好以及个人属性等。因此，从这些信息源中寻找标签的语义关联直观合理。

邻居相关信息源：用户的邻居相关的信息我们称作邻居相关信息源，例如邻居用户的标签、发布的微博等等。许多研究证明，一个用户更倾向于与具有相同兴趣、属性的人进行交互[18]。因此，从邻居信息源中识别语义关联也是合理可行的。

然而，从上述不同信息源中准确的定义标签的语义关联，面临以下两点挑战：

- 用户的上下文信息非常复杂且存在大量噪音。例如，每个用户可能发布大量的不同主题、不同形式的微博，并不是所有的微博都能够体现出用户标签的语义信息。
- 用户的上下文信息包含多个不同的异构信息源，每个信息源都有各自的特点。如何联合地对这些信息源进行建模具有重要意义。

为了解决上述问题，我们提出了一个概率生成模型，标签关联模型 (Tag Correspondence Model, TCM)，来探究用户的标签在不同来源之中的语义关联。对于每个信息源，我们选取一些语义元素作为关联的候选集合。以用户发布的微博为例，我们可以利用微博中的词或者短语来作为关联元素的候选集合。TCM 会迭代的学习每个关联元素在所有标签上的概率分布，也能够根据不同用户的特点，调整不同来源的关联元素的比重。

除了对标签与不同来源中的元素的关联关系进行建模，TCM 还能够对于未标注标签的用户进行标签推荐，来得到这些用户基于标签的显式表示。实验中，我们构建了一个真实的数据集，把用户标签推荐作为我们的评测任务。实验结果表明，TCM 显著的超过已有的标签推荐方法。

3.2　相关工作

标签推荐任务一直是社交媒体领域重要的研究问题。已有的研究工作主要关注如何对社交媒体中的 Web 页面、图片、视频等进行标签标注。

作为重要的个性化推荐任务，推荐系统领域的一些有效的技术也被引入来解决社会标签推荐问题，例如，基于用户-物体的协同过滤[47–49]，矩阵和张量分解[50–52]

等等。此外，一些基于图的算法也被用来解决社会标签推荐任务[53]。在这些基于图的方法中，需要依据用户历史的标注行为，构建一个基于用户-物体-标签的三分图，在这个基础之上，一些随机游走算法可以对标签进行排序。我们将上述这类方法归纳为基于协同的方法。

上述的研究主要依靠用户的历史标注行为来进行社会标签推荐。此外，还有一些工作关注如何利用元数据来推荐标签。这些方法我们称其为基于内容的方法。一些研究者将每个标签看作一个分类的类别，将标签推荐任务当做一个多标签分类任务来处理[54–59]。在这些方法中，特征与标签之间的语义关系隐藏在分类器的参数中，因此对于人来说是不可解释的。

受隐藏主题模型，例如 Latent Dirichlet Allocation (LDA)[60] 的启发，一些概率图模型的方法被提出，来对用户-物体-标签之间的关系进行建模。一个直观的想法是，假设标签和特征词都是由同样的隐藏主题来生成。通过将标签和特征词表示成隐藏主题的分布，从而可以可根据物体的元数据来推荐标签[30,61,62]。此外，Bundschus et al.[63] 对上述工作进行了扩展，提出了一个用户-标签-词的联合隐含主题模型。Content Relevance Model (CRM)[64] 也同样被提出，用来探究文本与标签之间的关系，用来进行标签推荐。与一些基于分类的方法和典型的基于文档-标签的主题模型方法 Corr-LDA[65] 相比，CRM 取得了最好的效果。

对于社交媒体用户标签推荐来说，用户只能够给自己进行标签标注，所以我们不能直接采用基于协同的的方法。此外，我们希望能够对用户标签的语义进行解释，所以基于分类的方法也不能够胜任。考虑到图模型方法的概率解释能力，我们提出了一个概率图模型，标签关联模型 TCM。尽管已经有一些图模型的标签推荐方法被提出，但是它们主要用来解决对标签和特定的因子之间的关系进行建模，例如用户、词语等，不能够对丰富的上下文信息进行同时建模。相对而言，TCM 能够考虑到多源异构的特征，建立起这些异构特征与标签之间的语义关联。

3.3　模型框架

在介绍模型之前，我们首先给出一些必要的符号定义。假设我们有一个微博用户集合 U。每个用户 $u \in U$ 会有一些相关的异构信息，例如个人简介、微博等等。此外，用户会给自己标注一个标签集合 $\vec{\boldsymbol{a}}_u$。标签集合中的每个标签都来自一个固定的标签词表 T，词表大小为 $|T|$。此外，用户还有有一个邻居用户的集合 $\vec{\boldsymbol{f}}_u$。

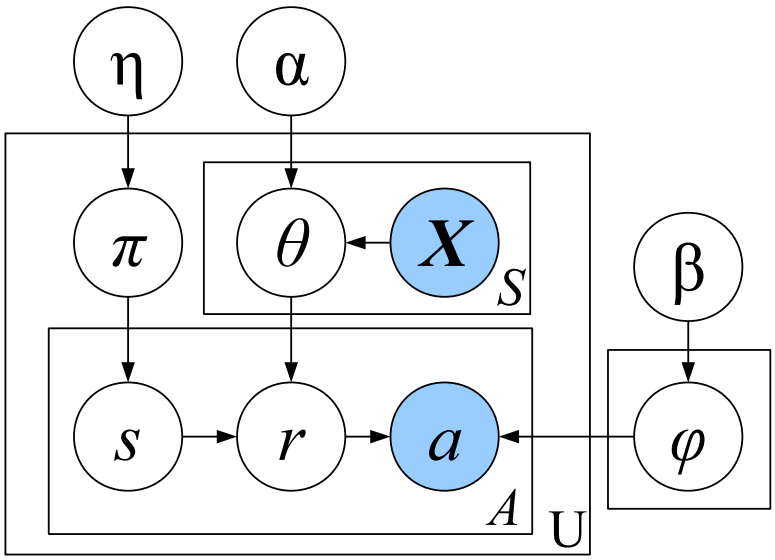


图 3.1　标签关联模型（Tag Correspondence Model）。

3.3.1　标签关联模型

我们提出标签关联模型 TCM，来识别用户的标签与多种来源信息之间的关联关系，这些信息来源并不局限于用户发布的微博、个人简介、邻居用户。

我们在图 3.1中，展示了 TCM 的概率图模型示意图。在 TCM 中，不失一般性，我们定义用户 u 所有的信息源集合为 S_u，其中每个信息源 $s \in S_u$ 可以表示成一个对于该来源词表 V_s 的权重向量 $\vec{\boldsymbol{x}}_{u,s}$。该词表中的所有元素都被看作标签的关联候选元素。其中的每个候选关联元素 r 可以表示成一个关于标签词表 T 中所有标签的多项分布 $\boldsymbol{\phi}_{s,r}$，该多项分布符合狄利克雷先验（图中的 β）。每个用户 u，都有一个来源的分布 $\boldsymbol{\pi}_u$，该分布符合狄利克雷先验 η_u。对于每个来源 s，针对当前来源词表 V_s 存在一个混合分布 $\boldsymbol{\theta}_{u,s}$，该分布符合狄利克雷先验 $\vec{\boldsymbol{\alpha}}_{u,s}$，表示对于当前用户该来源词表中不同关联元素的先验重要性。$\vec{\boldsymbol{x}}_{u,s}$ 表示归一化后的当前来源中不同关联元素的分布。每个关联元素 r 的先验为 $\alpha_{u,s,r} = \alpha x_{u,s,r}$，其中 α 为 LDA[66] 中人工预定义的参数。

在 TCM 中，每个用户 u 标注的标签 t 的生成过程如下：

- 根据分布 $\boldsymbol{\pi}_u$ 选取一个来源 s；
- 根据分布 $\boldsymbol{\theta}_{u,s}$ 选取一个当前来源中的关联元素 r;
- 根据分布 $\boldsymbol{\phi}_{s,r}$ 选取一个标签 t。

因此，标签 t 的选取，依赖于用户对于来源 s 的偏好程度，来源 s 对于关联元素 r 的偏好程度，以及关联元素 r 对于标签的偏好程度。

需要注意的是，我们定义一个来源为全局来源（*global source*），其中仅仅包含一个关联元素，而且该元素包含于所有的用户中。在本章工作中，我们假设每

一个标签都能够通过不同的来源进行解释。但实际上，有些流行的标签并不能通过用户的不同来源中的信息来解释，因此我们引入这个全局来源来处理这种情况。当我们不能给某个标签关联合适的来源或者元素时，我们会认为这个标签受到全局偏好的影响。

如 TCM 模型图3.1所示，用户标注的标签以及不同来源的关联元素是观测变量。我们需要找到一个有效的方式来计算观测变量的联合概率，以及隐藏变量的分配情况。联合概率形式化如下：

$$\Pr(\boldsymbol{a}, \boldsymbol{s}, \boldsymbol{r}|\boldsymbol{x}, \alpha, \eta, \beta) = \prod_{u \in U} Pr(\vec{a}_u, \vec{s}_u, \vec{r}_u|\vec{x}_u, \alpha, \eta, \beta).$$

给定用户 u，我们省略每个向量的下标，上述公式的右侧可以进一步形式化为：

$$\Pr(\vec{a}, \vec{s}, \vec{r}|\vec{x}, \alpha, \eta, \beta) = \Pr(\vec{a}|\vec{r}, \beta) \Pr(\vec{r}, \vec{s}|\vec{x}, \alpha, \eta).$$

通过优化上述联合概率，我们能够获得 TCM 模型中的参数，包括 π, θ 以及 ϕ。在上述联合概率中，第一项 $\Pr(\vec{\boldsymbol{a}}|\vec{\boldsymbol{r}}, \beta)$ 和 LDA 中词的生成过程类似，因此我们采用[66]中同样的推导。第二项可以分解为：

$$\Pr(\vec{r}, \vec{s}|\vec{x}, \alpha, \eta) = \Pr(\vec{r}|\vec{s}, \vec{x}, \alpha) \Pr(\vec{s}|\eta).$$

根据 Gregor[67] 中的公式 (52)，这两部分可以进一步形式化为：

$$\begin{aligned}
\Pr(\vec{s}|\vec{\eta}) &= \int_{\vec{\pi}} \Pr(\vec{s}|\vec{\pi}) \Pr(\vec{\pi}|\vec{\eta}) d\vec{\pi} \\
&= \int_{\vec{\pi}} \prod_{i=1}^{|\vec{x}|} (\mathbf{Multi}(s_i|\vec{\pi})) \mathbf{Dir}(\vec{\pi}|\vec{\eta}) d\vec{\pi} \\
&= \frac{\Delta(n_{u,:,\cdot,\cdot} + \vec{\eta})}{\Delta(\vec{\eta})},
\end{aligned}$$

以及

$$\begin{aligned}
\Pr(\vec{r}|\vec{s}, \vec{x}, \vec{\alpha}) &= \int_{\theta} \Pr(\vec{r}|\theta, \vec{s}) \Pr(\theta|\vec{x}, \vec{\alpha}) d\theta \\
&= \int_{\vec{\theta}} \prod_{i=1}^{|\vec{x}|} (\mathbf{Multi}(r_i|\theta_{s_i})) \prod_{i=1}^{|\vec{x}|} (\mathbf{Dir}(\vec{\theta_{x_i}}|\vec{\alpha})) d\vec{\theta}
\end{aligned}$$

$$= \prod_{s_i} \frac{\Delta(n_{u,s_i,:,\cdot} + \vec{\alpha}_{u,s_i})}{\Delta(\vec{\alpha}_{u,s_i})}.$$

这里，$\Delta(\vec{\alpha})$ 为 Gregor[67] 中的狄利克雷 δ 函数。$n_{u,j,k,t}$ 表示用户 u 中，标签 t 被分配到来源 s 中的关联元素 r 的次数。此外，我们利用符号 "$\cdot$'' 表示对这些频度的求和，以及用符号 ":'' 表示选取一个向量。

在 TCM 中，每个关联元素仅仅属于一个来源，因此我们不需要显式的使用来源$\vec{s}$。我们可以采用吉布斯采样算法[68] 来进行关联元素的分配以及参数估计。利用吉布斯采样更新的公式如下：

$$\begin{aligned}
&\Pr(s_{u,i} = j, r_{u,i} = k | \vec{s}_{\neg u,i}, \vec{r}_{\neg u,i}, a_{u,i} = t, \alpha, \beta, \eta) \\
&= \hat{p}_{(\neg u,i)}(a_{u,i} = t | r_{u,i} = k) \hat{p}_{(\neg u,i)}(r_{u,i} = k | s_{u,i} = j) \\
&\hat{p}_{(\neg u,i)}(s_{u,i} = j),
\end{aligned}$$

其中，上述公式的三个概率分别可以如下计算：

$$\begin{aligned}
\hat{p}_{(\neg u,i)}(s_{u,i} = j) = \hat{\pi}_{u,s} &= \frac{n^{(\neg u,i)}_{u,j,\cdot,\cdot} + (\vec{\alpha}_S)_j}{n^{(\neg u,i)}_{u,\cdot,\cdot,\cdot} + \sum_{j \in S} (\vec{\alpha}_S)_j}, \\
\hat{p}_{(\neg u,i)}(r_{u,i} = k | s_{u,i} = j) = \hat{\theta}_{u,s,r} &= \frac{n^{(\neg u,i)}_{u,j,k,\cdot} + \alpha_{u,j,k}}{n^{(\neg u,i)}_{u,j,\cdot,\cdot} + \alpha_{u,j,\cdot}}, \\
\hat{p}_{(\neg u,i)}(a_{u,i} = t | r_{u,i} = k) = \hat{\phi}_{s,r,t} &= \frac{n^{(\neg u,i)}_{\cdot,j,k,t} + \beta}{n^{(\neg u,i)}_{\cdot,j,k,\cdot} + |T|\beta}.
\end{aligned}$$

通过上述公式，我们可以得到：

$$\begin{aligned}
&\Pr(s_{u,i} = j, r_{u,i} = k | \vec{s}_{\neg u,i}, \vec{r}_{\neg u,i}, a_{u,i} = t, \alpha, \beta, \eta) \\
&= \frac{n^{(\neg u,i)}_{\cdot,j,k,t} + \beta}{n^{(\neg u,i)}_{\cdot,j,k,\cdot} + |T|\beta} \frac{n^{(\neg u,i)}_{u,j,k,\cdot} + \alpha_{u,j,k}}{n^{(\neg u,i)}_{u,j,\cdot,\cdot} + \alpha_{u,j,\cdot}} \frac{n^{(\neg u,i)}_{u,j,\cdot,\cdot} + (\vec{\alpha}_S)_j}{n^{(\neg u,i)}_{u,\cdot,\cdot,\cdot} + \sum_{j \in S} (\vec{\alpha}_S)_j} \\
&\propto \frac{n^{(\neg u,i)}_{\cdot,j,k,t} + \beta}{n^{(\neg u,i)}_{\cdot,j,k,\cdot} + |T|\beta} \left(n^{(\neg u,i)}_{u,j,k,\cdot} + \alpha x_{u,j,k} \right).
\end{aligned} \tag{3-1}$$

这里，$\neg u, i$ 表示当前的频度统计不包括当前的分配。为了简化，我们定义 $(\vec{\alpha}_S)_j = \alpha_{u,j,\cdot}$。因此，上述公式中的第二个分数的分子部分可以与第三个分数的分子部分抵消。此外，第二个分数的分母部分对于不同的来源和关联元素为常数，也可以

被忽略。这样，我们发现上述更新规则与 LDA 的更新规则类似。

我们可以根据公式 (3-1) 估计 TCM 中的隐藏参数。吉布斯采样过程中，对于每个标签的来源分配和关联分配通过不断采样得到，相应的频度统计也会一直更新。最终，我们根据收敛之后的分配情况来估计 TCM 中的参数，如下所示：

$$\pi_{u,s} = \frac{n_{u,s,\cdot,\cdot} + \eta}{n_{u,\cdot,\cdot,\cdot} + |S|\eta}, \tag{3-2}$$

$$\theta_{u,s,r} = \frac{n_{u,s,r,\cdot} + \alpha x_{u,s,r}}{n_{u,s,\cdot,\cdot} + \alpha x_{u,s,\cdot}}, \tag{3-3}$$

$$\phi_{s,r,t} = \frac{n_{\cdot,s,r,t} + \beta}{n_{\cdot,s,r,\cdot} + |T|\beta}. \tag{3-4}$$

3.3.2　用户标签推荐

完成 TCM 模型的训练之后，每个标签 t 的含义可以根据它对应的关联元素分配情况 $\phi_{s,r,t} = \Pr(r|t)$ 进行解释。接下来我们介绍如何利用估计好的参数进行用户标签推荐，也就是根据用户的来源信息来推导标签的分布。

给定一个用户 u 以及对应的来源 S。选取一个标签 t 的概率可以形式化为：

$$\Pr(t|u,\phi) = \sum_{s\in S}\sum_{r\in V_s} \Pr(t|r,\phi)\Pr(r|u,s)\Pr(s|u),$$

其中，$\Pr(r|u,s) = \theta_{u,s,r}$, $\Pr(t|r,\phi) = \phi_{s,r,t}$，$\Pr(s|u)$ 表示用户对于来源 s 的偏好程度。这里，我们利用全局的偏好来进行近似，也就是 $\Pr(s|u) = \Pr(s)$。为了计算 $\Pr(s)$，我们构建了一个验证集来评测利用每个单独的来源进行标签推荐的效果。我们把不同来源的效果（$M = 10$ 时的 F 值) 作为每个来源的权重，然后把归一化之后的权重作为 $\Pr(s)$。

最后，我们根据计算出来的概率 $\Pr(t|u,\phi)$ 来对标签进行排序，选取概率最大的一些标签作为推荐结果。

3.4　特征源选取

在这一小节中，我们详细介绍 TCM 模型中每一个信息源的选取以及对应关联元素的构建。此外，我们还定义了不同信息源中，关联元素的加权方式，来作为公式 (3-1) 计算$\vec{x}$ 的先验知识。

3.4.1　用户相关特征源

我们考虑以下两种用户相关的特征源，包括微博和个人介绍：

微博：　对于用户发布的每条微博，我们有多种构建关联元素的方法，例如词、隐藏主题等等。在本章工作中，我们直接采用词作为关联元素，每个词的权重计算由下面两个统计量决定：

- 用户 u 的微博中包含该词语的比例；
- 所有用户中使用该词语的比例。

受 TF-IDF[69] 的启发，我们定义对于每个词语 w 的微博频度和逆用户频率 (MF-IUF)，也就是 $\text{MF-IUF}_{u,w} = \frac{|M_{u,w}|}{|M_u|} \times \log \frac{|U|}{|U_w|}$。其中，$M_{u,w}$ 表示用户 u 发布的微博中包含词语 w 的集合；M_u 表示用户发布的微博集合；U_w 表示所有用户 U 中，发布的微博中包含 w 的用户集合。

个人介绍：　一个微博用户通常会为自己提供一个简短的描述。尽管这个描述长度有限，一般只有几十个词，但是包含了关于用户属性、兴趣的高质量的信息。和 TF-IDF 以及 MF-IUF 类似，我们定义 $\text{UF-IUF}_{u,w} = \frac{n_{u,w}}{n_{u,\cdot}} \times \log \frac{|U|}{|U_w|}$ 来计算个人介绍中每个词的权重，其中，$n_{u,w}$ 为用户 u 中个人介绍中使用词语 w 的频度，U_w 表示在个人介绍中使用词语 w 的用户集合。

3.4.2　邻居相关特征源

除了用户相关特征源之外，我们还考虑微博用户邻居的信息，来帮助用户的标签推荐，包括：邻居的标签和邻居的个人介绍。

已经存在许多工作，将网络结构信息引入概率图模型中，例如，NetSTM[70] 以及 RTM[71]。这些方法的基本想法是，对于相邻文档的主题分布进行平滑，使得邻居之间的主题分布尽可能相似。尽管这些方法为结合用户相关信息和邻居相关信息提供了有效的思路，但是仍然面临两个问题：

- 这些方法不能直观的对这些来源与主题或标签的对应关系进行建模；
- 当对一个文档进行建模时，这些方法会考虑所有邻居文档的文本内容以及最新的主题分布，会造成巨大的内存和计算开销。.

这里，我们采用了一种简单的方法来结合邻居相关的信息，也就是把邻居的不同信息，作为额外的特征源，这种方法的有效性也经过了实验验证。

邻居标签：　对于一个用户 u，它的邻居标签能够反映出当前用户在个人中心网络里面的属性或兴趣。因此，我们把邻居的标签作为关联元素，每个标签的权重受以下两个因素的影响：

- 邻居中拥有该标签的比例；
- 所有用户中拥有该标签的比例。

与 TF-IDF 类似，我们定义 NF-IUF 来衡量每个邻居标签的权重，也就是 $\text{NF-IUF}_{u,t} = \frac{|N_{u,t}|}{|N_{u,\cdot}|} \times \log \frac{|U|}{|U_t|}$，其中，$N_{u,t}$ 表示 u 的邻居中拥有该标签的用户集合，N_u 为用户 u 的邻居集合，U_t 为拥有标签 t 的所有用户集合。

邻居介绍： 和邻居标签类似，我们也使用邻居的个人介绍信息来作为邻居相关的信息源。每个词的权重计算方式和 TF-IDF 类似，定义为 $\text{NF-IUF}_{u,w} = \frac{|N_{u,w}|}{|N_u|} \times \log \frac{|U|}{|U_w|}$，其中 $N_{u,w}$ 表示邻居中在个人介绍中使用词 w 的用户集合，U_w 表示所有在个人介绍中使用词 w 的用户集合。

3.5　实验结果

3.5.1　数据集

我们从新浪微博随机爬取 2 百万用户，时间范围从 2012 年 1 月到 2012 年 12 月。由于新浪微博中的许多用户会随意的填写自己的个人信息，所以我们在该数据集上进行了筛选，来选择高质量的训练数据。我们从中选取了 341, 353 含有完整的个人信息、微博、社交网络的用户，并且这些用户都标注了 2 个及以上的标签。此外，我们同样筛选出 4, 126 个出现频度大于等于 500 的标签。根据我们的统计，这些高频标签占到了所有标注标签标注次数的 98.67%。平均下来，每个用户拥有 4.54 个标签，63.35 个邻居，305.24 个邻居标签。用户个人介绍平均包含 6.93 个词。

3.5.2　实验设置

在 TCM 中，我们遵循 LDA[66] 的设定，设置超参数 $\alpha = 10$，$\beta = 0.1$。

实验中，我们用 UM, UD, NT 和 ND 分别表示用户微博、用户个人介绍、邻居标签和邻居介绍四个特征源。

为了更直观的展示 TCM 模型的高效和有效性，我们分析了模型的收敛情况、标签的特点以及关联元素的特点。此外，在3.5.4节中，我们针对用户标签推荐任务进行了定量的评测。

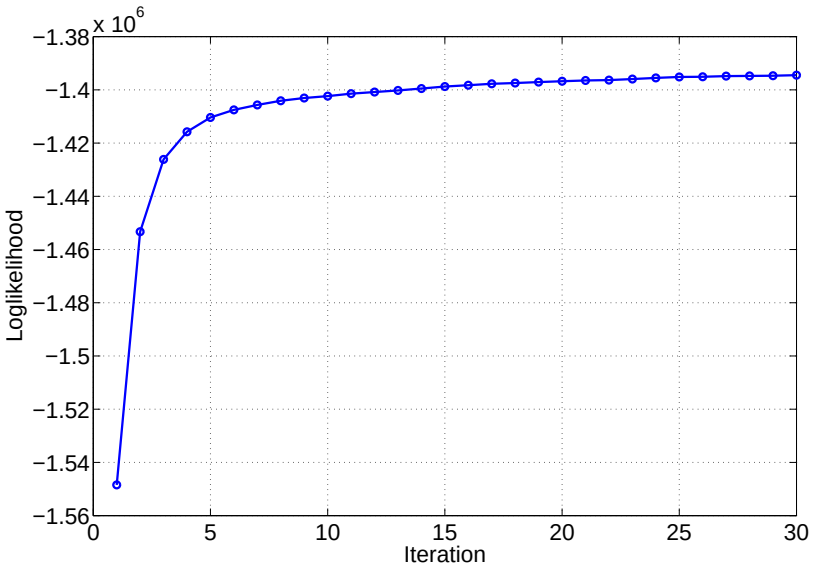


图 3.2　TCM 模型收敛情况。

3.5.3　经验性分析

3.5.3.1　模型收敛情况

图 3.2展示了模型训练过程中对数似然值的收敛趋势。这里的对数似然在是每一轮 TCM 训练完成后，对于测试集中的一部分计算的对数似然值，计算方式如下：

$$L(U_T) = \sum_{t \in U_T} \log \sum_{c,r} \Pr(t|c, r, \phi) \Pr(c, r|u).$$

我们发现，在迭代 15 轮左右，TCM 模型效果开始稳定，与传统的主题模型 LDA[60] 相比，收敛迅速，而且没有出现过拟合的情况。

3.5.3.2　不同信息源代表性标签

为了更好的理解四个特征源，在表3.1中，我们展示了不同特征源的权重 $\Pr(s)$，以及最有代表性的 5 个标签。这里，$\Pr(s)$ 直接通过对所有标签的分配情况求平均得到，也就是：

$$\Pr(s) = \frac{n_{\cdot,s,\cdot,\cdot} + \eta}{n_{\cdot,\cdot,\cdot,\cdot} + |S|\eta}.$$

为了选取每个来源代表性的标签，我们依照 Cohn et al.[72]，来如下计算每个标签 t 在每个来源 s 下的得分：

$$C(s,t) = \Pr(t|s) \times \Pr(s|t).$$

其中，

$$\Pr(t|s) = \frac{n_{\cdot,s,\cdot,t} + \beta}{n_{\cdot,s,\cdot,\cdot} + |T|\beta},$$
$$\Pr(s|t) = \frac{n_{\cdot,s,\cdot,t} + \beta}{n_{\cdot,\cdot,\cdot,t} + |S|\beta}.$$

表 3.1　不同特征源权重及代表性标签。

来源	$\Pr(s)$	Top 5 代表标签
UM	0.19	移动互联网, 方大同, 重庆, 深圳, 广州
UD	0.19	平面模特, 淘宝店主, 摄影师, cosplay, 电子商务
NT	0.42	网购, 小说, 媒体, 读书, 广告
ND	0.20	豆瓣, 懒, 小说, 美食, 音乐

从表3.1中，我们发现，邻居相关的特征源比用户相关的特征源权重更大。此外，邻居标签的权重达到 0.42，是最为重要的特征源。这是因为，一个用户自己产生的个人信息更加主观和随意，因此不能完全反映出用户的所有标签。然而，邻居的标签和描述，某种程度上可以看作其它用户对于当前用户的协同标注，因此更加可靠。

对于每个特征源的代表性标签，我们发现，邻居相关的信息源更能够反映出用户的兴趣，例如 ‘‘网购”, ‘‘读书’’，‘‘美食’’ 和 ‘‘音乐’’。与之相对应的是，用户信息相关的标签，更能够反映出用户的属性，例如，职业、地点和身份等。这些结果表明，属性标签可以在用户个人信息中找到比较好的关联关系，而兴趣标签可以从邻居信息中找到合适的关联关系。

此外，需要注意的是，TCM 对于全局信息源的设置也非常重要。全局信息源能够处理那些没有关联关系的标签。该信息源最有代表性的五个标签分别为 ‘‘音乐”, ‘‘电影”, ‘‘美食”, “80 后” 和 ‘‘旅行’’。这些标签非常流行和普遍，和用户的上下文信息没有太强的关联。如果不引入这种全局信息源，这些标签会错误的影响关联关系的识别。

3.5.3.3　不同标签代表性关联元素

表 3.2　李开复的标签的关联元素。

标签	Top 5 关联元素
教育	互联网 (NT), 教育 (UD), 教育 (UM), 政治 (NT), 学习 (NT)
科技	Android (NT), 互联网 (NT), 产品 (ND), 创新 (ND), 通信 (NT)
创业	创业 (NT), 风险投资 (NT), 电子商务 (NT), 企业家 (NT), 互联网 (UD)
移动互联网	SNS (NT), 移动 (UD), 互联网 (UM), 移动 (UM), IT (NT)
电子商务	B2C (NT), IT (NT), 电子商务 (UM), 电子商务 (NT), 市场 (NT)

TCM 的任务是为每个用户的标签找到对应的关联元素。这里，我们从李开复的标签中选取一些例子进行展示。在表3.2中，我们列出了这些标签最有代表性的一些关联因素，每个关联因素的权重通过 $C(r,t) = \Pr(t|r) \times \Pr(r|t)$ 计算得到。此外，我们还展示了每个关联元素对应的特征源。从该表中，我们发现，TCM 能够对于每个标签，从不同的特征源中识别中合理的关联元素。

3.5.4　实验评测

3.5.4.1　评测指标和基准方法

对于用户标签推荐任务，我们采用 precision (P), recall (R) 以及 F-Measure (F) 三个评测指标。给定一个微博用户，我们假设它实际的标签集合为 T_a，推荐的标签集合为 T_s，因此，准确预测的标签为 $T_s \cap T_a$。那么，上述三个评测指标的计算方式为：

$$P = \frac{T_s \cap T_a}{T_s}, R = \frac{T_s \cap T_a}{T_a}, F = \frac{2PR}{P+R}.$$

我们采用 5 层交叉验证，并且汇报平均的 P/R/F 的结果。实验中，每个用户推荐的标签数量 M 的范围为 $1-10$。

为了进行比较，我们选择 kNN[69]，TagLDA[62] 以及 NetSTM[70] 来作为基准方法。其中，kNN 根据训练集中最相似的 k 个用户来推荐标签。TagLDA 是一个代表性的隐藏主题模型，具体细节可参考 Si et al.[62]。在本文中，我们修改了原始的 NetSTM[70] 模型，来把标签当成主题，这样能够同时利用用户信息以及网络结构信息来推荐标签。我们设置 TagLDA 的主题数量 $K = 200$，kNN 的最近邻数量 $k = 5$，NetSTM 中的正则项系数 $\lambda = 0.15$。

此外，我们我们也测试了 TCM 模型只考虑单独一种特征源的效果，记为 TCM-UM/UD/NT/ND。TMC-UN 表示考虑所有特征源的方法。

3.5.4.2　实验结果和分析

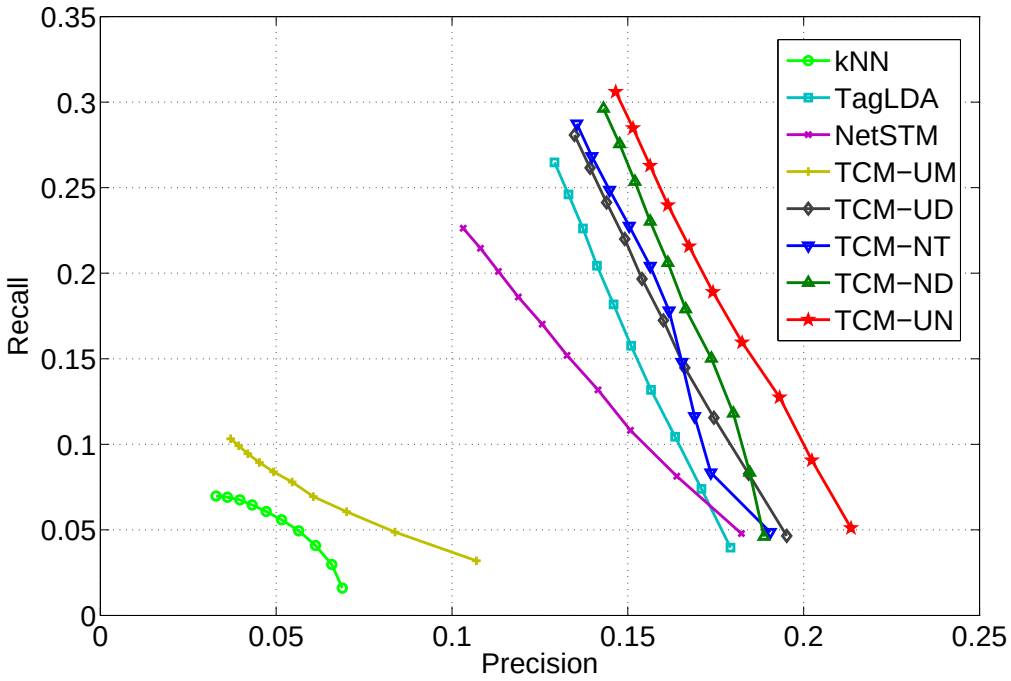


图 3.3　社会标签推荐结果。

在图3.3中，我们展示了不同方法的标签推荐的准确率/召回率曲线。曲线上的每个点表示推荐不同数量的标签时的准确率和召回率。当 $M = 1$ 时，点位于右下方，此时准确率较高而召回率很低。当 $M = 10$ 时，对应着左上角的点，此时召回率很高而准确率很低。曲线越靠近右上角，说明对应方法的效果越好。

从图3.3中，我们发现：

- 考虑所有特征源信息的 TCM 显著且一致的超过所有的基础方法。这验证了 TCM 结合多个特征源来进行标签推荐的有效性和合理性。
- 在基准方法中，*k*NN 和 Tag-LDA 只能够考虑一种用户相关的文本信息（个人介绍），从而效果较差。对于 *k*NN 来说，个人介绍往往太短，不足以用来计算用户之间的相似度。NetSTM 能够一定程度上引入网络结构信息，所以在推荐较多标签时，效果超过了 Tag-LDA 的方法。这也验证了网络结构信息对于标签推荐的重要性。

需要注意的是，从图 3.3中，我们发现 TCM 模型的绝对效果要低于与其它社

会标签推荐系统[51,62]。这主要由两方面原因造成：首先，由于微博用户只能够对自己进行标签标注，更为随意，而其它的标签推荐系统中，通常有成千上万的用户进行协同的标签标注。另外一方面，我们在评测的过程中进行严格的匹配，因此，即使一个方法能够给用户推荐合理的标签，但是由于没有被用户自身标注，所以也会被认为是错误标签。这种评测方法能够用来比较不同方法的效果，但不太适合评价一个方法实际的绝对效果。

此外，我们还探究了不同来源推荐的标签的重叠情况，如表3.3所示。对于每个特征源，我们展示了它们正确推荐的标签数量，以及与其它特征源同时推荐正确的标签比例。我们发现，不同特征源之间推荐正确的标签重叠比例较低，大多数低于 50%，这说明不同特征源关注的标签类型存在明显的差异，也验证了我们考虑不同的特征源进行标签推荐的合理性。

表 3.3　不同特征源标签推荐情况。

特征源	正确标签数量	UM	UD	NT	ND
UM	12,707	-	0.517	0.481	0.428
UD	16,191	0.406	-	0.593	0.403
NT	19,856	0.308	0.484	-	0.292
ND	16,038	0.339	0.407	0.362	-

3.5.4.3　示例

表 3.4　不同特征源对于李开复的标签推荐情况（加粗的标签表示推荐正确的标签）。

	推荐的 Top-5 标签
UM	**移动互联网**, **创业**, **互联网**, **电子商务**, 宅男
UD	**创新**, 自由, **互联网**, Google, **创业**
NT	**互联网**, 电影, **创业**, 旅行, **电子商务**
ND	**互联网**, **创业**, **电子商务**, 市场, **移动互联网**
UN	**创业**, **电子商务**, **互联网**, **移动互联网**, 读书

在表3.4中，我们展示了使用不同特征源的 TCM 模型推荐的 top-5 标签的情况。我们可以发现，不同特征源大部分推荐的标签都是正确的。尽管有些标签没有被李开复标注，例如 “Google”, ‘‘市场”, ‘‘旅行”, ‘‘电影”, ‘‘读书’’ 等，但是，根据它的个人信息判断，这些标签都有一定的相关性。这也表明，尽管标签推荐的绝对评测效果较低，但是并不表明这些方法推荐的标签不合理，质量较差，而是由于完全匹配的评测方式引起的。

3.6　本章小结

在本章工作中，我们提出利用主题标签来表示社交网络中的用户节点。由于大部分社交网络用户没有标签的标注信息，我们提出了一个概率生成模型 TCM，来对未标注标签的用户进行标签推荐。具体来说，TCM 能够探究标签与用户的不同的类型信息之间的关联关系。利用发现的这些关联关系，能够根据未标注用户的用户信息和邻居信息，推荐合理的标签。在真实数据集上的实验结果表明，TCM 模型显著地优于已有的标签推荐算法。

第 4 章　基于最大间隔的隐式网络表示

上述章节主要介绍了社会网络节点的显式表示方法，例如用词项、标签等来表示一个网络节点。虽然这些表示方式具有很好的可解释性，但是由于词项、标签等显式表示方法表示维度过高，会面临着计算效率的问题。为了提高网络节点表示的计算效率和在社会网络分析任务上的效果，我们探究利用表示学习的方法，来为网络中的节点学习低维实值的向量表示。该低维实值向量蕴含了网络节点的网络结构信息，也可以融合网络中丰富的异构信息，在计算效率得到提高的同时，在许多典型的网络分析任务上也取得了出色的效果。

然而，已有的网络表示学习方法一般都是无监督的方法，这些方法学习到的网络节点表示，往往缺乏区分性，在进一步的节点分类等预测任务中表现较差。在本章[①]中，我们为了克服该挑战，提出了一个半监督的基于最大间隔理论的网络表示学习模型，Max-Margin DeepWalk (MMDW)。MMDW 能够同时训练最大间隔分类器和网络表示学习模型。受最大间隔分类器的影响，学习到的网络节点表示不光包含节点的网络结构信息，也包含它们的类别标签信息，因此具有很好的区分性。在多个真实数据集的实验结果表明，我们提出的方法能够显著的提升节点分类的效果。

4.1　问题描述

网络表示在社会网络分析领域一直扮演着重要的角色。一个有效的网络表示有助于许多网络分析任务，例如节点分类、聚类、链接预测等等。作为网络中的基础的元素，一个网络节点通常被表示成一个离散的符号，也就是独热编码表示（one-hot representation）。这种表示方式直观简洁，在许多网络分析任务中有着广泛的应用。然而，独热编码表示通常会遇到数据稀疏性问题，不能够有效的考虑节点之间的关联关系或相似程度。

受近些年分布式表示学习的启发，网络表示学习的概念被研究学者提出来解决数据稀疏性的问题。网络表示学习会为每个节点学习一个低维实值的向量表示，来反映它的网络结构信息。这种表示适用于典型的网络分析任务，而且能够反映出节点之间的相关程度。近些年来，许多网络表示学习的方法不断被提出，例如 DeepWalk[1] 和 LINE[6]。DeepWalk 是一种基于节点局部信息的在线的网络表示学

① 本章主要工作以 “Max-Margin DeepWalk: Discriminative Learning of Network Representation’’ 为题发表在 2016 年的国际学术会议 “The International Joint Conference on Artificial Intelligence (IJCAI’16)’’ 上。

习模型。给定一个网络，它首先会进行随机游走，来得到节点序列。利用得到的大量的随机游走序列，DeepWalk 把每个节点看成词，把节点序列看成句子，采用典型的训练词向量的模型 Skip-Gram[5] 来训练节点表示。这种对于节点和词的类比非常直观有效，在多标签节点分类任务上取得了不错的效果。

然而，大部分已有的网络表示学习模型都是无监督的方法。尽管学习到的节点表示能够适用于不同的任务，但是它们在分类预测任务上的效果不够理想。值得指出的是，在实际世界的网络中存在着丰富的类别标签信息。例如下图 4.1所示，维基百科中的词条页面往往被标注了诸如“艺术”、“历史”、“科技”等标签。对于“TensorFlow”这个词条，它被标注了“应用机器学习”、“数据挖掘和机器学习软件”、“深度学习”等标签。在一些学术网络中，论文和作者一般也会被标注一些领域标签，来方便进行检索。这些标签类别信息对于网络节点的特征进行了高度的摘要概括，但是在网络表示学习模型中没有得到很好的利用。

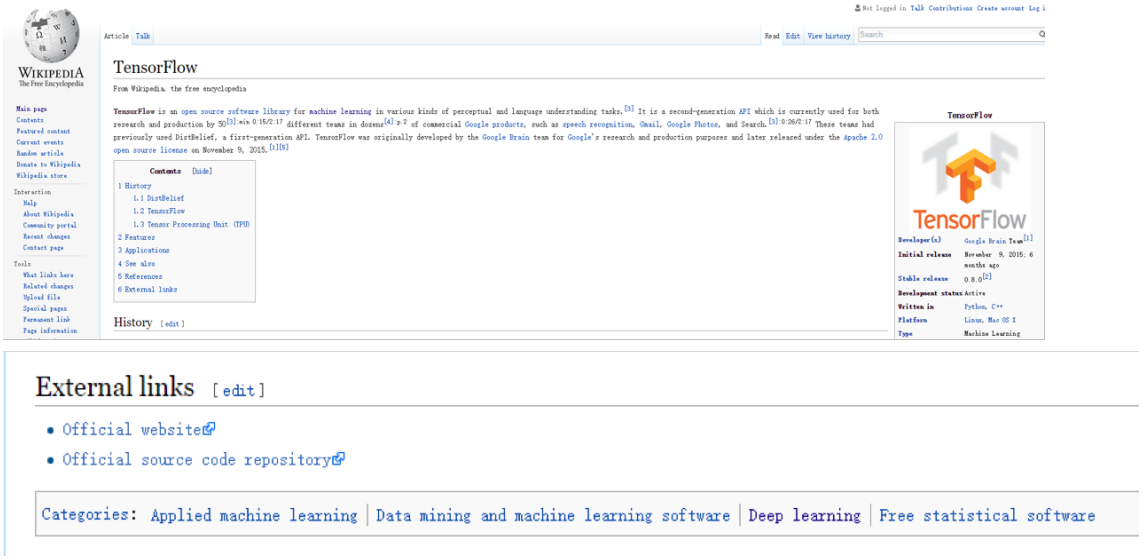


图 4.1　维基百科词条页面示例。

因此，我们想要探究如何在网络表示学习过程中充分利用节点的标签类别信息，来学习有区分性的网络节点表示。受最大间隔理论的启发，我们提出了基于最大间隔的 DeepWalk 模型（Max-Margin DeepWalk, MMDW），来为社会网络中的节点学习有区分性的网络表示，提高其在分类预测任务中的效果。如图 4.2所示，MMDW 首先会学习矩阵分解形式的 DeepWalk，随后，它会训练一个最大间隔分类器，例如支持向量机（Supprot Vector Machine, SVM）[73]。在训练得到的最大间隔分类器的基础上，MMDW 会尝试增大支持向量和分类边界之间的距离，从而使得不同类别的节点的表示更有区分性，也更加适用于一些分类预测任务。

总结来说，在这个工作中，我们有以下三个主要的创新点：

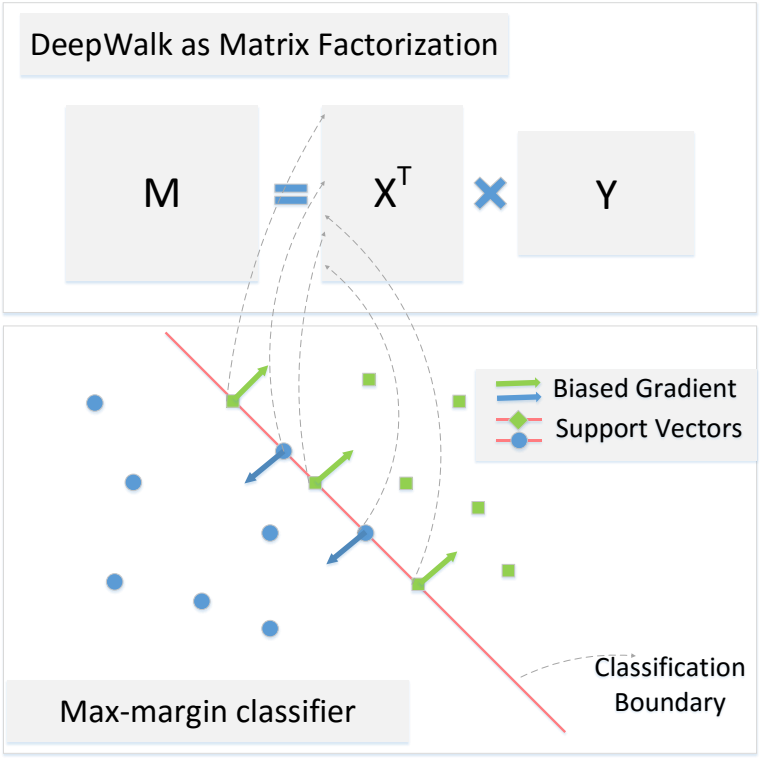


图 4.2　MMDW 模型示意图。

(1) 我们提出了一个有区分性的网络表示学习模型，MMDW，来将节点的类别标签信息融入节点表示向量中。MMDW 是首先利用最大间隔理论的半监督的网络表示学习模型。

(2) 我们提出了网络表示学习中的偏置梯度（Biased Gradient）的概念。一个向量的偏置梯度是指它应该偏置或移动的方向。这种偏置能够增大两个类别节点之间的距离，在梯度下降算法中作为额外的梯度被考虑进来。

(3) 我们在多个真实的数据集上进行了节点分类的实验，来验证 MMDW 的效果。实验结果表明，MMDW 比已有的网络表示学习方法有着显著而且一致的提升（5% 到 10%）。此外，我们利用 t-SNE 对于学习到的网络节点表示进行了可视化，来证明 MMDW 学习到的节点表示更有区分性。

4.2　相关工作

网络表示学习目的是为社会网络中的每个节点学习一个低维实值的向量表示。最有代表性的网络表示学习模型是 DeepWalk 和 LINE。受自然语言处理中用来训练词向量的 Skip-Gram[5] 模型的启发，Perozzi et al.[1] 提出了 DeepWalk 模型，通过将节点当做词，节点构成的随机游走序列当做句子，来学习网络节点表示。LINE[6]

通过对网络节点中的一阶邻近度和二阶邻近度进行建模，来学习大规模网络的节点表示。

此外，在其它领域中，最大间隔理论得到了许多应用。[74] 首次在马尔科夫网络中引入最大间隔理论。 Zhu et al.[75] 提出了最大熵判别的 LDA[60]（Maximum Entropy Discrimination LDA, MedLDA），来学习一个有区分性的主题模型。此外，在自然语言处理领域的典型任务中，最大间隔理论也有许多应用，例如，分词[76] 和语义分析[77] 。

然而，在网络表示学习领域，还没有工作可以有效利用节点的类别标签信息。为了解决该问题，我们提出了 MMDW 来为社会网络中的节点学习有区分性的网络表示。

4.3　模型框架

在这一节中，我们介绍半监督的网络表示学习模型 MMDW。MMDW 是一个基于矩阵分解的网络表示学习模型，该模型会同时优化最大间隔分类器 SVM 和矩阵分解形式的网络表示学习模型。

4.3.1　问题定义

假设存在一个网络 $G = (V, E)$，其中 V 是节点集合，E 是边的集合，也就是 $E \subset V \times V$。网络表示学习模型目的是为每个节点 $v \in V$ 学习一个低维实值的表示向量 $\mathbf{x}_v \in \mathbb{R}^k$，其中，$k$ 是表示空间的维度，而且 $k << |V|$。学习到的节点表示包含了该节点在网络中的角色信息，可以用来度量节点之间的相关程度，也可以在分类任务中作为特征向量。给定一个标签 $l \in \{1, \cdots, m\}$，我们可以根据节点的表示向量训练逻辑回归或者 SVM 分类器。这里，m 是标签的数量。

在接下来的部分，我们会先介绍典型的网络表示学习模型 DeepWalk，以及它对应的矩阵分解形式。之后，我们会详细介绍我们提出的 MMDW 模型。

4.3.2　矩阵分解形式 DeepWalk

DeepWalk[1] 首先会在网络中进行随机游走，来得到节点的随机游走序列。随后，它采用训练词向量的 Skip-Gram[5] 模型，来学习节点表示。

受 SKip-Gram 启发，DeepWalk 最大化目标节点和它随机游走窗口中的上下文节点之间的共现概率。假设我们有一个随机游走序列 $\mathbf{s} = \{v_1, \ldots, v_s\}$。我们设置窗口大小为 K，因此对于每个节点 v_i, 它的上下文节点为 $\mathbf{c}_i = (v_{i-K}, \ldots, v_{i+K}) \setminus v_i$。从

而，DeepWalk 的目标函数为：

$$\mathcal{L}(S) = \sum_{\mathbf{s}\in S}[\frac{1}{M}\sum_{i=K}^{M-K}\sum_{v_j\in\mathbf{c}_i}\log \Pr(v_j|v_i)]. \tag{4-1}$$

这里，S 是所有生成的随机游走序列的集合。概率 $\Pr(v_j|v_i)$ 可以通过 softmax 函数计算得到：

$$\Pr(v_j|v_i) = \frac{\exp(\mathbf{x}_j \cdot \mathbf{x}_i)}{\sum_{t\in V}\exp(\mathbf{x}_t \cdot \mathbf{x}_i)}, \tag{4-2}$$

其中，$\mathbf{x}_j$ 和 $\mathbf{x}_i$ 分别是节点 v_j 和 v_i 的表示向量，$(\cdot)$ 是向量之间的点积。

针对 DeepWalk 模型，Yang et al.[78] 证明 DeepWalk 等价于对一个矩阵 M 进行矩阵分解。矩阵中的每一个元素为：

$$M_{ij} = \log \frac{[e_i(A + A^2 + \cdots + A^t)]_j}{t}. \tag{4-3}$$

其中，A 是网络 G 的转移矩阵，也就是网络的邻接矩阵的行归一之后的矩阵。e_i 是一个指示向量，其中第 i 个元素是 1，其余为 0。M_{ij} 表示节点 v_i 通过 t 步随机游走到节点 v_j 的平均概率的对数。

从公式 4-3我们发现，精确的计算矩阵 M 代价较高。因此，我们采用 Yang et al.[78] 中类似的设定，近似矩阵 $M = (A + A^2)/2$。我们没有取对数，是因为对数之后的矩阵包含大量的非零元素，会大大增加矩阵分解的计算开销[79].

最后，我们利用 $M = X^TY$ 对 DeepWalk 进行矩阵分解，通过两个矩阵 $X \in \mathbb{R}^{k\times|V|}$ 和 $Y \in \mathbb{R}^{k\times|V|}$ 来最小化如下的目标函数：

$$\min_{X,Y}\mathcal{L}_{DW} = \min_{X,Y}||M - (X^TY)||_2^2 + \frac{\lambda}{2}(||X||_2^2 + ||Y||_2^2), \tag{4-4}$$

其中，超参数 λ 控制正则项的权重。

4.3.3　最大间隔 DeepWalk

最大间隔方法，例如 SVM[73]，通常被用来解决不同的识别问题，例如文档分类和手写识别。

在该工作中，我们把学习到的网络表示 X 当做特征向量，来训练一个 SVM 节点分类器。假设训练集为 $\mathcal{T} = \{(\mathbf{x}_1, l_1), \cdots, (\mathbf{x}_T, l_T)\}$，多类别 SVM 通过解决下面的

带约束的线性优化问题来寻找一个最优的线性分类函数：

$$\begin{aligned} \min_{W,\xi} \mathcal{L}_{SVM} &= \min_{W,\xi} \frac{1}{2} \|W\|_2^2 + C \sum_{i=1}^{T} \xi_i \\ \text{s.t.} \quad & \mathbf{w}_{l_i}^T \mathbf{x}_i - \mathbf{w}_j^T \mathbf{x}_i \geq e_i^j - \xi_i, \quad \forall i, j \end{aligned} \tag{4-5}$$

其中，

$$e_i^j = \begin{cases} 1, & \text{if} \quad l_i \neq j, \\ 0, & \text{if} \quad l_i = j. \end{cases} \tag{4-6}$$

这里，$W = [\mathbf{w}_1, \cdots, \mathbf{w}_m]^T$ 为 SVM 的权重矩阵，$\xi = [\xi_1, \cdots, \xi_T]$ 是控制训练集中分类错误的松弛变量。

如上文所述，这种方法并不能够影响学习到的节点表示。给定学习到的网络节点表示，SVM 仅仅能够帮助找到一个最优的分类边界，这种情况下，本身学习到的节点表示并没有很好的区分性。

受基于最大间隔理论的主题模型 MedLDA[75] 的启发，我们提出同时优化上述的基于矩阵分解形式的 DeepWalk 模型以及最大间隔 SVM 分类器，来学习有区分性的节点表示。MMDW 的优化目标如下：

$$\begin{aligned} \min_{X,Y,W,\xi} \mathcal{L} &= \min_{X,Y,W,\xi} \mathcal{L}_{DW} + \frac{1}{2} \|W\|_2^2 + C \sum_{i=1}^{T} \xi_i \\ \text{s.t.} \quad & \mathbf{w}_{l_i}^T \mathbf{x}_i - \mathbf{w}_j^T \mathbf{x}_i \geq e_i^j - \xi_i, \quad \forall i, j \end{aligned} \tag{4-7}$$

4.3.4　优化算法

目标函数4-7中的训练参数包括节点表示矩阵 X，上下文表示矩阵 Y，SVM 的分类权重矩阵 W 以及松弛向量 ξ。为了优化 MMDW，我们设计了一个有效的优化算法，来迭代的对这些参数进行分别优化。通过引入偏置向量，矩阵分解过程会显著的受到最大间隔分类器的影响。

我们的优化算法包含下面两个步骤：

4.3.4.1　优化 W 和 ξ

当固定 **X** 和 **Y** 时，对于 MMDW 的优化变成一个标准的多类别 SVM 问题[80]。它的对偶形式如下：

$$\begin{aligned} \min_{\mathbf{Z}} \frac{1}{2}\|W\|_2^2 + \sum_{i=1}^{T}\sum_{j=1}^{m} e_i^j z_i^j \\ \text{s.t.} \quad \sum_{j=1}^{m} z_i^j = 0, \quad \forall i \\ z_i^j \le C_{l_i}^j, \quad \forall i, j \end{aligned} \tag{4-8}$$

其中，

$$\mathbf{w}_j = \sum_{i=1}^{l} z_i^j \mathbf{x}_i, \quad \forall j$$

以及

$$C_{y_i}^m = \begin{cases} 0, & \text{if} \quad y_i \neq m, \\ C, & \text{if} \quad y_i = m. \end{cases}$$

这里，我们将拉格朗日乘子 α_i^j 替代为 $C_{l_i}^j - z_i^j$。

为了求解该对偶问题，我们使用坐标下降方法，将 Z 拆分成 $[\mathbf{z}_1, \cdots, \mathbf{z}_T]$，其中

$$\mathbf{z}_i = [z_i^1, \cdots, z_i^m]^T, \quad i = 1, \cdots, T.$$

我们采用一个有效的序列对偶方法[81] 来解决 $\mathbf{z}_i$ 对应的子问题。

4.3.4.2　优化 X 和 Y

当 W 和 ξ 固定时，原问题 4-7变成求解带约束的矩阵分解问题，如下所示：

$$\begin{aligned} \min_{X,Y} \mathcal{L}_{DW}(X, Y; M, \lambda) \\ \text{s.t.} \quad \mathbf{w}_{l_i}^T \mathbf{x}_i - \mathbf{w}_j^T \mathbf{x}_i \ge e_i^j - \xi_i, \quad \forall i, j \end{aligned} \tag{4-9}$$

当不考虑约束条件时，我们可以计算出如下梯度：

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial X} &= \lambda X - Y(M - X^T Y),\\ \frac{\partial \mathcal{L}}{\partial Y} &= \lambda Y - X(M - X^T Y).\end{aligned} \tag{4-10}$$

$\forall i \in \mathcal{T}, j \in 1, \cdots, m$，如果 $l_i \neq j$ and $\alpha_i^j \neq 0$，根据 KKT 条件，我们可以得到

$$\mathbf{w}_{l_i}^T \mathbf{x}_i - \mathbf{w}_j^T \mathbf{x}_i = e_i^j - \xi_i. \tag{4-11}$$

当分类边界固定时，我们希望对于这些支持向量 $\mathbf{x}_i$ 进行偏置，使它们向自己真实的类别方向移动，来获得更准确的分类结果。通过对于分类边界上的支持向量进行移动，能够增大两个类别之间的区分性。

接下来我们解释这些偏置是如何计算的。给定一个训练集中的节点 $i \in \mathcal{T}$，对于第 j 个约束条件，我们对其表示向量 $\mathbf{x}_i$ 添加 $\alpha_i^j(\mathbf{w}_{l_i} - \mathbf{w}_j)^T$，这样，该约束条件变为：

$$\begin{aligned}&(\mathbf{w}_{l_i} - \mathbf{w}_j)^T(\mathbf{x}_i + \alpha_i^j(\mathbf{w}_{l_i} - \mathbf{w}_j))\\ &= (\mathbf{w}_{l_i} - \mathbf{w}_j)^T \mathbf{x}_i + \alpha_i^j ||(\mathbf{w}_{l_i} - \mathbf{w}_j||_2^2\\ &> e_i^j - \xi_i.\end{aligned} \tag{4-12}$$

需要注意的是，我们利用拉格朗日乘子 α_i^j 来判断一个节点是不是处在分类边界上。这样，只有对应着 $\alpha_i^j \neq 0$ 的向量 $\mathbf{x}_i$ 会被添加一个偏置。

对于节点 $i \in \mathcal{T}$，它的梯度变成

$$\frac{\partial \mathcal{L}}{\partial \mathbf{x}_i} + \eta \sum_{j=1}^{m} \alpha_i^j (\mathbf{w}_{l_i} - \mathbf{w}_j)^T, \tag{4-13}$$

也就是我们提到的偏置梯度。这里，η 来控制偏置向量和原始梯度之间的权重。

在 X 更新之前，W 和 ξ 满足 SVM 的 KKT 条件，这种情况下对应的解是最优解。然而，更新 X 之后，KKT 条件不再满足，会导致目标函数的轻微上升，但是实验发现，这种上升通常会在可接受的范围之内。

4.4　实验结果

在这一节中，我们通过节点分类实验来评测我们提出的模型的效果。此外，我们还对学习到的节点表示进行了可视化，来验证 MMDW 能够学习到更有区分性的网络节点表示。

4.4.1　数据集和实验设置

针对节点分类任务，我们采用了如下三个真实的网络数据集：

- **Cora.** Cora①是一个由 McCallum et al.[82] 构建的论文数据集，它包含 2,708 篇机器学习论文，这些论文被分成了 7 类。论文与论文之间的引用网络构成了一个典型的社会网络。
- **Citeseer.** Citeseer 是另外一个由 McCallum et al.[82] 构建的论文数据集。该数据集包含 3,312 论文以及 4,732 条它们之间引用的边。这些论文被分为了 6 类。
- **Wiki.** Wiki[83] 是一个网页数据集。该数据集包含了 2,405 个来自维基百科的 web 页面，其中页面之间的 17,981 个超链接关系构成了一个社会网络。这些页面被划分为 19 类。与上述两个数据集相比，该数据集更加稠密。

为了进行评测，我们从标注的节点中进行随机划分，来分成训练集和测试集。我们把训练比例从 10% 提高到 90%，来观察不同模型在不同比例下的节点分类效果。分类器方面，我们采用了 SVM[80] 来构建节点分类器。

4.4.2　基准方法

- **DeepWalk.** DeepWalk[1] 是一个典型的利用随机游走和词向量表示学习模型来学习节点表示的方法。对于 DeepWalk，我们设置窗口大小 $K = 5$，每个节点的游走序列长度 $\gamma = 80$，序列数量为 10。节点的表示维度 $k = 200$。
- **DeepWalk as Matrix Factorization.** 在上述章节中提到， Yang et al.[78] 证明 DeepWalk 等价于对一个矩阵 M 进行矩阵分解。我们采用该论文中的近似方法，使得 $M = (A + A^2)/2$，将优化得到的矩阵 X 作为节点的表示向量。
- **2nd-LINE.** LINE[6] 通过对节点之间的一阶邻近度和二阶邻近度进行建模，来学习大规模社会网络的节点表示。由于一阶邻近度只能对于无向网络进行建模，这里，我们采用二阶邻近度的方法 LINE (2nd-LINE) 来学习有向网络的节点表示。

① https://people.cs.umass.edu/ mccallum/data.html

4.4.3　实验结果和分析

表 4.1　Cora 数据集节点分类准确率 (%)。

%Labeled Nodes	10%	20%	30%	40%	50%	60%	70%	80%	90%
DW	68.51	73.73	76.87	78.64	81.35	82.47	84.31	85.58	85.61
MFDW	71.43	76.91	78.20	80.28	81.35	82.47	84.44	83.33	87.09
LINE	65.13	70.17	72.2	72.92	73.45	75.67	75.25	76.78	79.34
MMDW(10^{-2})	**74.94**	**80.83**	**82.83**	**83.68**	**84.71**	**85.51**	**87.01**	**87.27**	**88.19**
MMDW(10^{-3})	74.20	79.92	81.13	82.29	83.83	84.62	86.03	85.96	87.82
MMDW(10^{-4})	73.66	79.15	80.12	81.31	82.52	83.90	85.54	85.95	87.82

表 4.2　Citeseer 数据集节点分类准确率 (%)。

%Labeled Nodes	10%	20%	30%	40%	50%	60%	70%	80%	90%
DW	49.09	55.96	60.65	63.97	65.42	67.29	66.80	66.82	63.91
MFDW	50.54	54.47	57.02	57.19	58.60	59.18	59.17	59.03	55.35
LINE	39.82	46.83	49.02	50.65	53.77	54.2	53.87	54.67	53.82
MMDW(10^{-2})	**55.60**	60.97	63.18	65.08	**66.93**	**69.52**	**70.47**	**70.87**	**70.95**
MMDW(10^{-3})	55.56	**61.54**	**63.36**	**65.18**	66.45	69.37	68.84	70.25	69.73
MMDW(10^{-4})	54.52	58.49	59.25	60.70	61.62	61.78	63.24	61.84	60.25

表 4.3　Wiki 数据集节点分类准确率 (%).

%Labeled Nodes	10%	20%	30%	40%	50%	60%	70%	80%	90%
DW	52.03	54.62	59.80	60.29	61.26	65.41	65.84	66.53	68.16
MFDW	56.40	60.28	61.90	63.39	62.59	62.87	64.45	62.71	61.63
LINE	52.17	53.62	57.81	57.26	58.94	62.46	62.24	66.74	67.35
MMDW(10^{-2})	**57.76**	**62.34**	**65.76**	**67.31**	**67.33**	**68.97**	**70.12**	**72.82**	**74.29**
MMDW(10^{-3})	54.31	58.69	61.24	62.63	63.18	63.58	65.28	64.83	64.08
MMDW(10^{-4})	53.98	57.48	60.10	61.94	62.18	62.36	63.21	62.29	63.67

在表格 4.1, 4.2和4.3中，我们展示了不同方法在不同比例训练数据下的节点分类准确率。在这些表格中, 我们将 DeepWalk 缩写为 DW，矩阵分解形式的 DeepWalk 缩写为 MFDW，2nd-LINE 缩写为 LINE。此外，我们展示了不同 η 设置下的 MMDW 的实验结果。从这些表格中，我们观察发现：

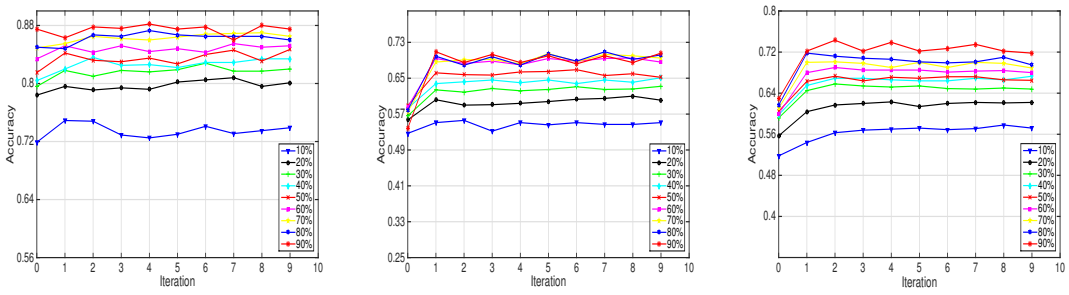
- 我们提出的 MMDW 模型在不同数据集以及不同训练比例下，显著且一致的优于所有的基准方法。其中，当训练比例为 50% 时，MMDW 在 Citeseer 数据集上获得了近 10% 的绝对提升，在 Wiki 数据集上获得了近 5% 的绝对提升。

DeepWalk 不能够对于这些数据集中的节点进行有效的节点表示，而 MMDW 能够很好的解决该问题。这些实验上的显著提升表明，我们提出的 MMDW 模型更加鲁棒，将最大间隔理论以及标签类别信息引入网络表示学习模型，能够有效的提升网络节点表示的质量。

- 值得注意的是，在 Citeseer 和 Wiki 数据集中，MMDW 在仅用一半训练数据的情况下，效果也优于原始的 DeepWalk 模型。这些结果表明 MMDW 更有潜力处理训练数据缺失的问题，更适合用来进行分类、预测任务。
- 我们提出的 MMDW 模型与传统的网络表示学习模型相比，获得了显著且一致的提升。与之形成对比的是，基准方法在稀疏程度不同的数据集上表现不稳定。这表明，引入标签类别的监督信息，使得 MMDW 能够有效的处理更多样的网络数据。

上述观察结果表明，MMDW 能够有效的结合节点的标签类别信息，生成高质量的网络节点表示。同时，MMDW 具有很好的鲁棒性和可扩展性。学习到的网络节点表示能够直接应用于更多的网络分析任务，包括节点相似度计算、链接预测等等。此外，偏置向量的想法也能够直接的应用于其它基于矩阵分解的模型。

4.4.4　收敛情况



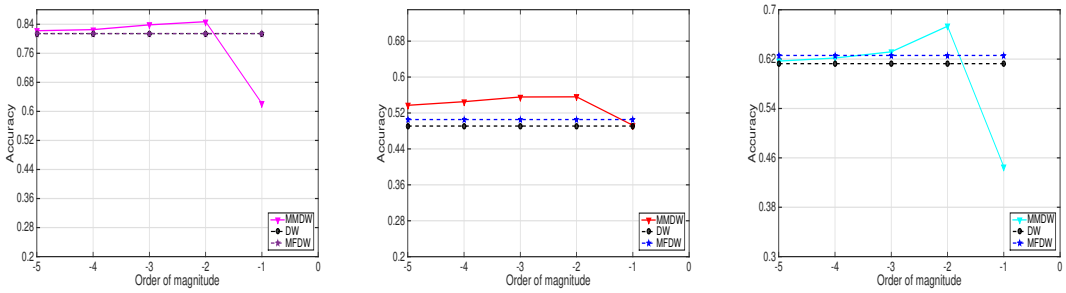
(a) Cora　　(b) Citeseer　　(c) Wiki

图 4.3　MMDW 在不同数据集上的收敛情况。

MMDW 迭代的优化最大间隔分类器和矩阵分解模型。为了探究 MMDW 在训练过程中的收敛情况，我们在图 4.3中展示了 MMDW 在不同数据集以及不同训练比例下，模型的效果随着训练轮次的变化情况。从这三个图中，我们发现，MMDW 在初始的几轮效果会持续上升，通常能够在训练 2 到 3 轮之后就能够达到最好的效果，随后 MMDW 的效果会逐渐稳定。这些结果表明，MMDW 在不同数据集及训练比例下都能够快速收敛，同时，由于 MMDW 中网络表示学习模型和最大间隔分类器能够同时影响节点的表示，随着训练轮数的增多，MMDW 也不会出现过拟

合的现象。上述观察再一次验证了我们提出的 MMDW 模型的鲁棒性和有效性。

4.4.5　参数敏感性分析



(a) Cora　　(b) Citeseer　　(c) Wiki

图 4.4　MMDW 在不同数据集上的参数敏感性分析。

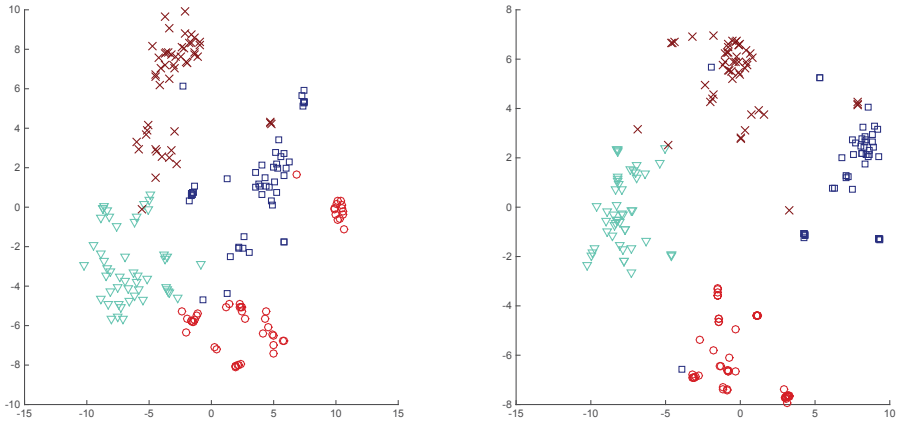


图 4.5　节点表示的 t-SNE 可视化 (左边：DeepWalk 右边：MMDW)。

在 MMDW 中，原始的 X 的梯度和偏置梯度通过不同的方式计算得到，它们初始时处于不同的量级。因此，我们引入参数 η 来控制和平衡偏置梯度和原始梯度的比重。为了观察该参数对于模型效果的影响，我们固定训练比例为 50%，观察 MMDW 在不同参数设置下的效果变化情况。

从图 4.4中，我们发现，当 η 的取值处于 10^{-5} 到 10^{-2} 之间时，MMDW 有着稳定的表现，而且这个参数取值范围适用于不同的三个数据集。当 η 取值过小时，此时 MMDW 几乎不受到最大间隔分类器的影响，模型的效果接近于原始的 DeepWalk 模型，而当 η 取值过大时，MMDW 中的网络表示学习部分对于节点表示的影响变弱，会使得网络节点的表示质量大幅下降。

表 4.4　Top-5 最近邻结果。

DeepWalk	
题目	类别
Truncating temporal differences On the efficient implementation of TD for reinforcement learning	Reinforcement Learning
Living in a partially structured environment How to bypass the limitation of classical reinforcement techniques	Neural Networks
Why experimentation can be better than perfect guidance	Theory
Averaged reward reinforcement learning applied to fuzzy rule tuning	Reinforcement Learning
A neural network pole balancer that learns and operates on a real robot in real time	Neural Networks
MMDW	
题目	类别
Applying online search to reinforcement learning	Reinforcement Learning
The efficient learning of multiple task sequences	Reinforcement Learning
A modular Q learning architecture for manipulator task decomposition	Reinforcement Learning
Truncating temporal differences On the efficient implementation of TD for reinforcement learning	Reinforcement Learning
Exploration and model building in mobile robot domains	Reinforcement Learning

4.4.6　节点表示可视化

在本章工作中，我们提出 MMDW 来学习有区分性的网络节点表示。为了验证学习到的节点表示是否更具有区分性，我们利用 t-SNE 来展示不同模型学习到的节点表示的可视化结果。如图 4.5所示，我们展示 DeepWalk 和 MMDW 两个模型在 Wiki 数据集上的可视化结果。其中，每个点表示测试集中的一个节点，每种颜色表示该节点对应的实际的类别。我们随机选取 4 个不同的类别来更清晰的展示不同类别之间的边界情况。

从该图中，我们发现，MMDW 能够学习到更好的聚类效果，不同类别之间的节点分类边界非常明显。相反的是，DeepWalk 模型学习到的节点表示倾向于混淆在一起，没有明显的分类边界。一个更好的划分的节点表示意味着更有区分性，在分类任务中更好进行预测。这些结果再一次验证了 MMDW 能够通过融合节点的类别标签信息，来学习到区分性的节点表示。

4.4.7　示例

我们从 Cora 数据集中选取了一个示例来更直观的展示 MMDW 的有效性。这篇选取的论文题目是 “Fast Online Q(λ)”, 该论文节点所属的类别为 “Reinforcement Learning’’。如表 4.4所示，我们列出了根据 MMDW 和 DeepWalk 学习到的表示计算出的 Top-5 最近邻结果。这里，我们用余弦距离来度量两个节点之间的相似度。

从该表中，我们发现，DeepWalk 计算出来的最近邻中，仅仅有 2 个邻居与该样例拥有共同的类别，与此对应的是，MMDW 找到的 5 个最近邻都属于同样的类别。根据该样例的题目，我们发现它是一篇关于 “Online Learning’’ 和 “Q-problem’’ 的文章。DeepWalk 找出的邻居大部分与这些主题没有关系，而 MMDW 发现的邻居与这两个题目非常相关。这个例子也再次表明，MMDW 考虑了节点的类别标签信息，提高了节点表示的质量。

4.5　本章小结

在本章工作中，我们针对已有网络表示学习工作不能有效的考虑节点的类别标签信息，从而导致节点表示区分性差的问题，提出了基于最大间隔理论的网络表示学习模型 MMDW。通过引入偏置梯度，该模型能够同时训练基于矩阵分解形式的网络表示模型和最大间隔分类器，最后使得学习到的节点表示更有区分性，有利于提高在预测任务上的效果。我们在三个真实世界的数据集上进行了节点分类实验，结果表明，MMDW 能够有效的提高节点分类的效果。此外，针对网络节点表示的可视化结果也验证了 MMDW 学习到的节点表示更有区分性。

第 5 章　上下文相关隐式网络表示

在上一章节中，我们介绍了在网络表示学习过程中，考虑节点的类别标签信息，来提高网络节点表示的区分性以及节点分类的效果。本章将针对网络分析的另外一个重要的任务，链接预测，来探究如何让网络表示学习能够更好的进行链接预测的任务。

已有的网络表示学习一般是为每个网络节点学习一个固定的低维实值的向量表示，也就是上下文无关的向量表示。然而，这种方式忽略了节点在与不同邻居节点交互时的多样的角色和特点。在这一章的工作[①]中，我们假设一个节点在与不同的邻居节点进行交互时，往往会展现出不同的方面，从而拥有相应的不同的节点表示。在这个工作中，我们提出了上下文相关的网络表示学习模型（Context-Aware Network Embedding, CANE）来解决该问题。通过考虑网络节点附加的文本信息和引入相互注意力机制，CANE 能够对节点之间的关系进行更准确的建模，从而学习上下文相关的节点表示。为了验证 CANE 的效果，我们在三个真实社会网络数据集上进行了实验，结果表明，CANE 比已有的上下文无关的网络表示学习模型在链接预测任务上取得了显著的提升。

5.1　问题描述

网络嵌入 (Network embedding, NE)，也就是网络表示学习 (network representation learning, NRL)，目的是将网络中的节点映射到低维的表示空间。网络表示学习能够解决传统基于符号的网络表示面临的稀疏性和计算效率问题，为处理目前大规模的社会网络提供了高效的解决途径。

在真实世界的社会网络中，一个节点在跟不同的邻居节点交互时，往往会展现不同的方面的特点。如图 5.1所示，一个研究者往往会跟不同的研究者在不同的研究题目下进行合作。具体来说，左侧的研究者（A）是一个自然语言处理领域句法分析方面的专家，他与中间的研究者 (B) 一起合作发表论文，因此在 A 的眼中，B 是一个句法分析方面的专家。然而，右侧的用户（C）是一个机器翻译方面的研究者，她也与 B 合作发表论文，那么在 C 的眼中，B 是一个机器翻译方面的专家。虽然 A 和 C 都与 B 有合作关系，但 B 在他们眼中的形象是完全不同的。这种情况在其它社会网络中也非常常见，例如社交媒体中的用户往往会与不同的朋友分享

① 本章主要工作以 “CANE: Context-Aware Network Embedding for Relation Modeling’’ 为题发表在 2017 年的国际学术会议 “The Annual Meeting of the Association for Computational Linguistics (ACL’17)’’ 上。

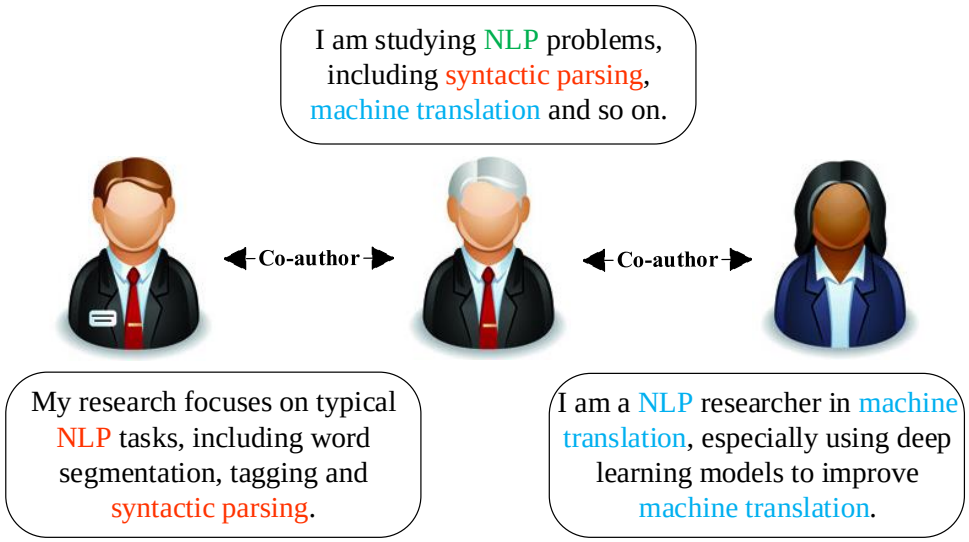


图 5.1　基于文本的信息网络示例。(红色、蓝色和绿色字体分别代表左边、右边以及两者共同关注的信息)。

不同的兴趣；一个 web 页面会因为不同的原因或目的链接到不同的页面。然而，目前已有的网络表示学习方法仅仅能够针对每个节点学习一个固定的节点表示，因此这些方面面临着以下两个问题：

- 已有的网络表示学习方法不能够处理节点在与不同邻居交互时角色转变的问题。
- 已有的网络表示学习方法倾向于使一个节点所有的邻居的表示向量尽可能相似，往往不符合真实场景。例如图 5.1中的左右两个用户，尽管他们几乎没有共同研究题目，但是由于他们都与中间的用户存在链接关系，从而会学习到相似的表示向量。

为了解决上述问题，我们提出了上下文相关网络表示学习模型，CANE，来对节点之间的交互关系进行更准确的建模。具体来说，CANE 能够处理包含外部信息的社会网络。在该网络中，每个节点蕴含着丰富的外部信息，例如文本、标签以及其它的元数据，在这种场景下，节点的上下文对于网络表示的帮助更加显著。不失一般性，我们在基于文本的信息网络上，验证 CANE 的效果。

在传统的网络表示学习模型中，每个节点被表示成一个静态的表示向量，也就是上下文无关表示（context-free embedding）。与之相反的是，CANE 会根据节点交互邻居的不同，为其分配动态的向量表示，也就是上下文相关表示（context-aware embedding）。以节点 u 和它的邻居节点 v 为例，u 的上下文无关节点表示会保持不变，无论 u 与哪个邻居进行交互。行成对比的是，u 的上下文相关网络表示会根据邻居的不同而发生变化。

当 u 与 v 发生交互时，我们根据两者的文本信息推断出它们互相关注的上下

文表示。对于每个节点，我们能够方便的采用已有的神经网络模型，例如卷积神经网络（convolutional neural networks, CNNs[84–86]）和循环神经网络（recurrent neural networks, RNNs[87,88]），来构建上下文无关的文本表示。为了实现上下文相关的文本表示，我们在神经网络中引入选择注意力机制，来发现节点 u 和 v 之间的互相注意力。互相注意力机制能够指导神经网络模型从文本信息中找出邻居节点关注的信息，从而学习到上下文相关的文本表示。

我们在三个真实世界的网络数据集上进行了实验。链接预测的实验结果验证了 CANE 模型的有效性。与已有方法相比，CANE 能够更准确的对节点之间的链接关系进行建模，因此在该任务上取得了显著的提升。除此之外，CANE 在节点分类任务上也取得了与最先进方法可比的结果，这也再次验证了我们提出的模型的适应性。

5.2　相关工作

近些年来，随着大规模社交网络的发展，网络表示学习被提了出来，用来解决大规模网络的分析任务。例如，DeepWalk[1] 利用随机游走策略得到网络中的节点序列，然后采用训练词向量的模型 Skip-Gram[5] 来学习网络节点表示。LINE[6] 利用一阶邻近度和二阶邻近度，优化节点之间的联合概率和条件概率，来学习大规模网络的节点表示。Node2vec[7] 改进了 DeepWalk 中的随机游走的策略，通过 BFS 和 DFS 来探究不同层次的网络结构。然而，这些方法仅仅能够考虑节点的网络结构信息，而忽略了真实的网络中节点丰富的异构信息。

为了解决该问题，研究者提出了一系列工作来融合社会网络中节点的异构信息。例如， Yang et al.[10] 提出了 Text-Associated DeepWalk (TADW)，来利用文本信息提高矩阵分解形式的 DeepWalk 的效果。 Tu et al.[89] 提出了最大间隔网络表示学习模型 Max-Margin DeepWalk (MMDW)，来利用节点的标签类别信息学习有区分新的网络节点表示。Chen et al.[90] 提出了 Group-Enhanced Network Embedding (GENE)，通过引入团体信息来提高网络表示的效果。Sun et al.[91] 通过将文本当成一种特殊的节点，提出了 Context-Enhanced Network Embedding (CENE)，来同时利用网络结构信息和文本信息学习网络节点表示。

然而，目前已有的网络表示学习方法只能够学习上下文无关的网络节点表示，忽略节点与邻居交互时的不同角色。因此，我们假设节点在与不同邻居交互时拥有不同的表示，提出了 CANE 来学习上下文相关的网络节点表示。

5.3　模型框架

5.3.1　问题定义

假设存在一个信息网络 $G = (V, E, T)$，其中 V 是节点集合，$E \subseteq V \times V$ 是节点之间边的集合，T 是节点对应的文本信息的集合。每条边 $e_{u,v} \in E$ 表示两个节点 (u, v) 之间的关系，$w_{u,v}$ 表示该边上的权重。此外，每个节点 $v \in V$ 的文本信息表示为一个词的序列 $S_v = (w_1, w_2, \ldots, w_{n_v})$，其中，$n_v = |S_v|$ 表示该序列的长度。信息网络表示学习的目的是为该网络中的每个节点 $v \in V$ 学习一个低维实值的向量表示，也就是 $\mathbf{v} \in \mathbb{R}^d$。该表示向量蕴含了节点的网络结构信息和其它附加信息，例如文本、标签等。这里，$d \ll |V|$ 为表示空间的维度。

定义 1 上下文无关网络表示（Context-free Embeddings）：传统的网络表示学习模型对于每个节点学习上下文无关的网络表示。也就是说，每个节点的表示向量是固定的，不会随着它的上下文信息（例如，交互的邻居节点）而改变。

定义 2 上下文相关网络表示（Context-aware Embeddings）：与传统的学习上下文无关表示的网络表示学习模型不同的是，CANE 能够对一个节点，根据它的上下文的不同，学习不同的表示向量。具体来说，对于一条边 $e_{u,v}$，CANE 能够学习上下文相关表示向量 $\mathbf{v}_{(u)}$ 和 $\mathbf{u}_{(v)}$。

5.3.2　模型目标

为了有效的利用网络节点的网络结构信息和附加的文本信息，我们提出了对于同一个节点 v 的两种类型的表示向量，也就是基于结构的表示向量 $\mathbf{v}^s$ 以及基于文本的表示向量 $\mathbf{v}^t$。基于结构的表示向量能够捕捉节点的网络结构信息，而基于文本的表示向量能够捕捉节点附加的文本信息。给定节点的这两种表示，我们可以简单的对这两者进行拼接，得到节点最终的表示 $\mathbf{v} = \mathbf{v}^s \oplus \mathbf{v}^t$，其中，$\oplus$ 代表向量拼接操作。需要注意的是，基于文本的表示向量 $\mathbf{v}^t$ 既可以是上下文相关的，也可以是上下文无关的，具体计算方式会在之后的小节进行介绍。当文本表示向量 $\mathbf{v}^t$ 是上下文相关的表示时，拼接得到的最终的表示向量 $\mathbf{v}$ 也同样是上下文相关的表示。

给定上述的定义，CANE 目的是最小化如下的目标函数：

$$\mathcal{L} = \sum_{e \in E} L(e). \tag{5-1}$$

这里，每条边的目标函数 $L(e)$ 包含如下两个部分：

$$L(e) = L_s(e) + L_t(e), \tag{5-2}$$

其中，$L_s(e)$ 表示基于结构的目标函数，$L_t(e)$ 代表基于文本的目标函数。

在接下来的小节中，我们会分别对这两部分目标函数进行详细的介绍。

5.3.3　基于结构的目标函数

不失一般性，我们假设 G 是一个有向的网络，因为对于无向网络来说，每条无向边可以转化成两条有同样权重、方向相反的有向边。

因此，基于结构的目标函数目的是利用基于结构的网络表示来衡量一条有向边的对数似然值：

$$L_s(e) = w_{u,v} \log p(\mathbf{v}^s|\mathbf{u}^s). \tag{5-3}$$

依照 LINE[6] 的做法，我们如下定义公式 (5-3) 中 v 指向 u 的条件概率：

$$p(\mathbf{v}^s|\mathbf{u}^s) = \frac{\exp(\mathbf{u}^s \cdot \mathbf{v}^s)}{\sum_{z \in V} \exp(\mathbf{u}^s \cdot \mathbf{z}^s)}. \tag{5-4}$$

5.3.4　基于文本的目标函数

真实世界中的网络节点往往伴随着文本信息。因此，我们利用这些文本信息，提出基于文本的目标函数，来学习节点的文本表示。

基于文本的目标函数 $L_t(e)$ 可以被设计成多种形式。为了和基于结构的目标函数 $L_s(e)$ 保持一致，我们如下定义 $L_t(e)$：

$$L_t(e) = \alpha \cdot L_{tt}(e) + \beta \cdot L_{ts}(e) + \gamma \cdot L_{st}(e), \tag{5-5}$$

其中，α、β 和 γ 控制不同部分的权重。公式 (5-5) 中不同部分的定义如下：

$$\begin{aligned} L_{tt}(e) &= w_{u,v} \log p(\mathbf{v}^t|\mathbf{u}^t), \\ L_{ts}(e) &= w_{u,v} \log p(\mathbf{v}^t|\mathbf{u}^s), \\ L_{st}(e) &= w_{u,v} \log p(\mathbf{v}^s|\mathbf{u}^t). \end{aligned} \tag{5-6}$$

公式 (5-6) 中的条件概率保证了节点的两种类型的表示（基于结构的表示和基于文本的表示）在统一的表示空间，具有一致性，也会考虑到两者各自的特点，不会强制使它们完全相等。同样的, 我们采用 softmax 函数来计算这些条件概率，如公式 (5-4) 所示。

同以往的网络表示学习方法一样，CANE 中基于结构的节点表示被当做模型的参数。但是对于基于文本的节点表示，我们希望通过节点附加的文本信息得到它。此外，基于文本的表示向量可以通过上下文无关或者上下文相关的方式得到。在下面的小节，我们会分别详细介绍这两种方式。

5.3.5　上下文无关的文本表示

在自然语言处理领域，已经存在很多神经网络模型，通过词的序列来得到对应的文本表示，例如卷积神经网络 CNN[84–86] 和循环神经网络 RNN[87,88]。

在本文中，我们探究了不同的文本建模的神经网络方法，包括 CNN、Bidirectional RNN[92] 以及 GRU[93]。最后，我们采用了表现最好的 CNN 来对文本进行建模。与其它方法相比，CNN 能够捕捉到词之间的局部的语义依赖关系。

将一个节点对应的词序列作为输入，CNN 通过三层网络来得到基于文本的表示向量，包括：查表层（Looking-up）、卷积层和池化层。

查表：给定一个词序列 $S = (w_1, w_2, \ldots, w_n)$，查表层将该序列中的每个词 $w_i \in S$ 通过查表操作，转化成其对应的词向量 $\mathbf{w}_i \in \mathbb{R}^{d'}$，最终得到该序列对应的词向量序列 $\mathbf{S} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$。这里，$d'$ 表示词向量的维度。

卷积： 给定词向量序列 $\mathbf{S}$，卷积层根据该序列抽取出词向量之间局部的特征。具体来说，针对一个长度为 l 的滑动窗口，卷积层利用一个卷积矩阵 $\mathbf{C} \in \mathbb{R}^{d \times (l \times d')}$ 来计算后续的特征，如下所示：

$$\mathbf{x}_i = \mathbf{C} \cdot \mathbf{S}_{i:i+l-1} + \mathbf{b}, \tag{5-7}$$

其中，$\mathbf{S}_{i:i+l-1}$ 表示第 i 个滑动窗口中词向量的拼接向量，$\mathbf{b}$ 是偏置向量。需要注意的是，我们对于序列边缘的卷积窗口进行了零向量填充[94]。

池化： 为了得到节点的文本表示向量 $\mathbf{v}^t$，我们针对卷积的结果 $\{\mathbf{x}_0^i, \ldots, \mathbf{x}_n^i\}$ 进行最大池化操作和非线性变换操作，如下所示：

$$r_i = \tanh(\max(\mathbf{x}_0^i, \ldots, \mathbf{x}_n^i)), \tag{5-8}$$

最后，我们通过对节点对应的文本信息进行 CNN 编码，得到了节点对应的文

本表示向量 $\mathbf{v}^t = [r_1, \ldots, r_d]^T$。根据上述计算过程，$\mathbf{v}^t$ 只与该节点对应的文本信息有关，与节点交互的邻居无关，因此我们称之为上下文无关的文本表示。

5.3.6　上下文相关的文本表示

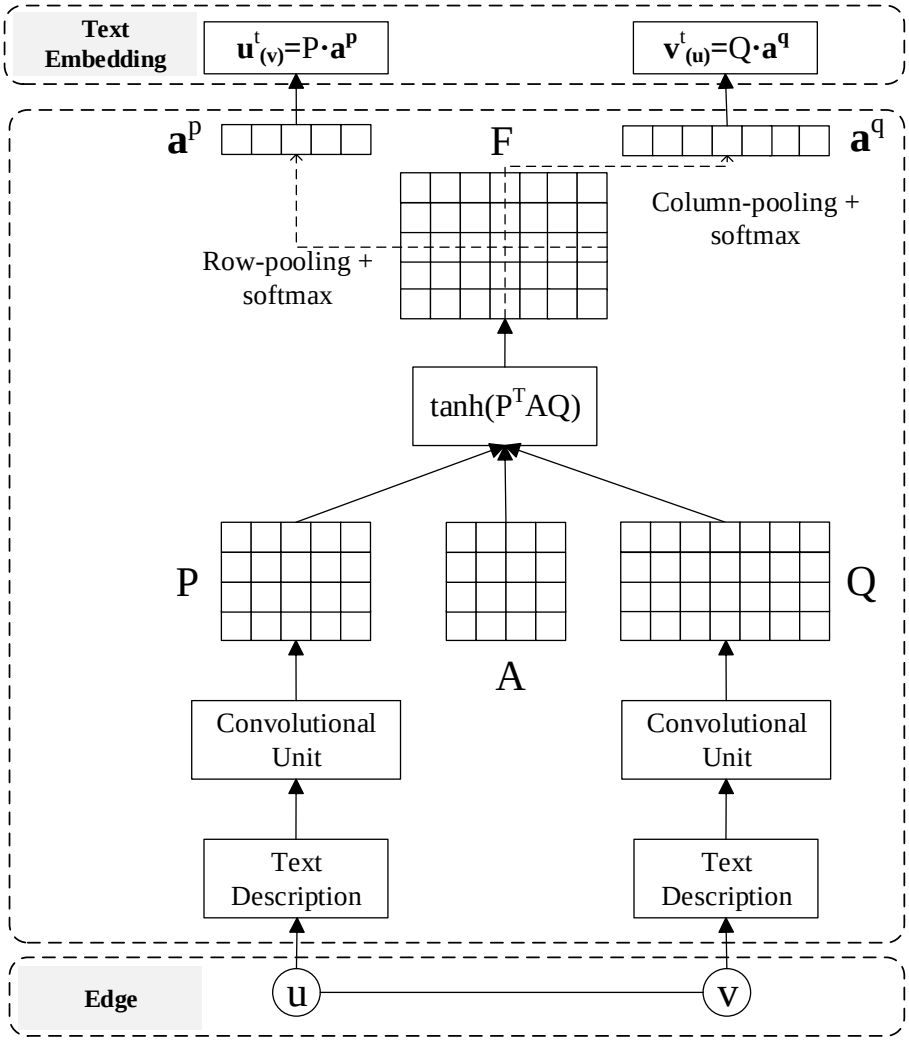


图 5.2　上下文相关文本表示示意图。

如前面所述，在 CANE 中，我们假设一个特定的节点往往在与不同邻居交互的过程中扮演着不同的角色。也就是说，每个节点应该对于不同的邻居节点的文本信息有不同的关注点，也就是上下文相关文本表示。

具体来说，我们采用了互相注意力机制（mutual attention）来获得上下文相关的文本表示。该机制能够使得 CNN 中的池化层与一条边上的两个节点相关，也就是说，使得一个节点的文本信息能够影响另外一个节点的文本表示，反之亦然。

在图 5.2中，我们给出了上下文相关文本表示向量的生成过程。对于一条边 $e_{u,v}$

以及对应的两个文本序列 S_u 和 S_v，我们能够通过卷积操作得到两个序列对应的卷积结果 $\mathbf{P} \in \mathbb{R}^{d \times m}$ 和 $\mathbf{Q} \in \mathbb{R}^{d \times n}$。这里，$m$ 和 n 分别代表两个文本序列 S_u 和 S_v 的长度。通过引入一个注意力矩阵 $\mathbf{A} \in \mathbb{R}^{d \times d}$，我们计算得到两个文本序列之间的关联矩阵 $\mathbf{F} \in \mathbb{R}^{m \times n}$，如下所示：

$$\mathbf{F} = \tanh(\mathbf{P}^T \mathbf{A} \mathbf{Q}). \tag{5-9}$$

其中，关联矩阵中的每个元素 $\mathbf{F}_{i,j}$ 代表两个序列中的隐向量 $\mathbf{P}_i$ 和 $\mathbf{Q}_j$ 之间的相关程度。

随后，我们对关联矩阵 $\mathbf{F}$ 分别进行沿行的池化和沿列的池化（行池化和列池化），来得到权重向量。实验中，我们发现平均池化（mean-pooling）的效果优于最大池化（max-pooling），因此，我们采用如下的平均池化操作来得到每一个权重：

$$\begin{aligned} g_i^p &= \mathbf{mean}(\mathbf{F}_{i,1}, \ldots, \mathbf{F}_{i,n}), \\ g_i^q &= \mathbf{mean}(\mathbf{F}_{1,i}, \ldots, \mathbf{F}_{m,i}). \end{aligned} \tag{5-10}$$

矩阵 $\mathbf{P}$ 和 $\mathbf{Q}$ 的权重向量分别为 $\mathbf{g}^p = [g_1^p, \ldots, g_m^p]^T$ 和 $\mathbf{g}^q = [g_1^q, \ldots, g_n^q]^T$。

接下来，我们采用 softmax 函数，来将上述权重向量进行归一，得到最终的注意力向量 $\mathbf{g}^p$ 和 $\mathbf{g}^q$。例如，$\mathbf{a}^p$ 的第 i 个元素为：

$$a_i^p = \frac{\exp(g_i^p)}{\sum_{j \in [1,m]} \exp(g_j^p)}. \tag{5-11}$$

最终，节点 u 和 v 的上下文相关文本表示向量为：

$$\begin{aligned} \mathbf{u}_{(v)}^t &= \mathbf{P}\mathbf{a}^p, \\ \mathbf{v}_{(u)}^t &= \mathbf{Q}\mathbf{a}^q. \end{aligned} \tag{5-12}$$

因为两个节点的文本表示向量在生成过程中互相相关，所以我们称上述文本表示向量为上下文相关的文本表示向量。

给定一条边 (u, v)，我们通过对基于结构的表示向量和上下文相关的文本表示向量进行拼接，得到两个节点上下文相关的表示向量 $\mathbf{u}_{(v)} = \mathbf{u}^s \oplus \mathbf{u}_{(v)}^t$ 和 $\mathbf{v}_{(u)} = \mathbf{v}^s \oplus \mathbf{v}_{(u)}^t$。

5.3.7　优化算法

根据公式 (5-3) 和公式 (5-6)，CANE 目标是优化 $\mathbf{u} \in \{\mathbf{u}^s, \mathbf{u}^t_{(v)}\}$ 和 $\mathbf{v} \in \{\mathbf{v}^s, \mathbf{v}^t_{(u)}\}$ 之间的条件概率。然而，直接利用 softmax 函数来优化该条件概率计算开销较大。因此，我们采用负采样算法（negative sampling）[4] 来将每个条件概率转化成如下形式：

$$\log \sigma(\mathbf{u}^T \cdot \mathbf{v}) + \sum_{i=1}^{k} E_{z \sim P(v)}[\log \sigma(-\mathbf{u}^T \cdot \mathbf{z})], \tag{5-13}$$

其中，k 表示负例的数量，σ 表示 sigmoid 函数。$P(v) \propto {d_v}^{3/4}$ 表示节点的分布，d_v 表示节点 v 的出度。

然后，我们采用 Adam[95] 优化算法来优化基于负采样的目标函数。需要注意的是，CANE 非常适合零样本（zero-shot）的场景，对于没有邻居的新节点，CANE 可以利用训练好的 CNN 来生成它的文本表示向量。

5.4　实验结果

为了验证 CANE 对于节点之间关系建模的有效性，我们在多个真实的数据集上进行的链接预测实验。此外，我们同样进行了节点分类实验，来验证上下文相关的节点表示能否组成高质量的上下文无关的节点表示。

5.4.1　数据集

表 5.1　数据集统计信息。

数据集	**Cora**	**HepTh**	**Zhihu**
节点	2,277	1,038	10,000
边	5,214	1,990	43,894
类别	7	–	–

我们选取了如下三个网络数据集进行实验：

- **Cora**[①]是由 McCallum et al.[82] 构建的典型的学术论文引用网络。过滤掉该数据集中不包含文本信息的节点之后，剩下 2,277 篇机器学习相关的论文，这些论文被分成了 7 类。

① https://people.cs.umass.edu/~mccallum/data.html

- **HepTh**① (High Energy Physics Theory) 是由 Leskovec et al.[96] 公开的来自 arXiv②的高能物理学术论文数据集。我们同样过滤掉了不包含摘要信息的论文，最终保留 1,038 篇论文。
- **Zhihu**③是中国最大的在线问答社区。在该网络中，用户会互相关注，并且回答感兴趣的问题。我们随机抓取了 10,000 个活跃的用户以及它们之间的关注关系，并且把他们关注的话题的文本描述作为文本信息。

关于这三个数据集的具体统计信息如表 5.1所示。

5.4.2　基准方法

我们采用了如下两类方法作为基准方法，包括仅仅基于结构的传统的网络表示学习模型，以及同时考虑网络结构和文本信息的网络表示学习模型。

基于结构的方法：

- *MMB*[97] (Mixed Membership Stochastic Blockmodel) 是一个典型的针对关联数据的图模型。在该生成模型中，当生成每条边时，每个节点会随机选取一个不同的主题。
- *DeepWalk*[1] 在网络上进行随机游走来获得节点序列，然后采用训练词向量的 Skip-Gram 模型[5] 来学习网络节点表示。
- *LINE*[6] 对节点之间的一阶邻近度和二阶邻近度进行建模，来学习大规模网络中的网络节点表示。
- *Node2vec*[7] 针对 DeepWalk 模型中的随机游走策略进行改进，利用 BFS 和 DFS 搜索算法，来获取不同层次的网络结构信息。

基于结构和文本的方法：

- *Naive Combination:* 我们将表现最好的基于结构的网络表示和基于 CNN 的文本表示进行简单的拼接，来得到节点的表示向量。
- *TADW*[10] 采用矩阵分解的形式，来结合节点的文本信息和网络结构信息，学习信息网络节点的表示。
- *CENE*[91] 通过将文本当作特殊类型的节点，来同时利用网络中的结构信息和文本信息，对于不同节点之间的边设计条件概率，学习网络节点的表示。

① https://snap.stanford.edu/data/cit-HepTh.html
② https://arxiv.org/
③ https://www.zhihu.com/

5.4.3　评测指标和实验设置

对于链接预测任务，我们采用一个标准的评测指标 **AUC**[98]。该指标表示存在的边上两个节点之间的相似度大于不存在的边上两个节点相似度的概率。

对于节点分类任务，我们采用 L2 正则的逻辑回归分类器 (L2R-LR)[42] 来训练节点分类器。此外，我们采用准确率来评价不同模型节点分类的效果。

为了公平起见，我们将所有模型表示向量的维度设置为 200。除了 CANE，所有模型中负采样算法的负例数量为 5。对于 LINE 方法，我们对于一阶邻近度和二阶邻近度，分别学习 100 维的表示向量，最后拼接成 200 维的表示向量。在 nodevec 和 CANE 中，我们采用网格搜索的方法，来选择表现最好的超参数。此外，为了提高训练速度，我们将 CANE 中的负例数量 k 设置为 1。为了展示考虑互相注意力机制和公式 (5-3)、(5-6) 中两种类型目标函数的有效性，我们设计了三种版本的 CANE 模型，包括只考虑文本表示的 CANE with text only，不考虑注意力机制的 CANE without attention 以及完整的 CANE。

表 5.2　Cora 数据集链接预测结果 ($\alpha = 1.0, \beta = 0.3, \gamma = 0.3$)。

% 保留的边比例	15%	25%	35%	45%	55%	65%	75%	85%	95%
MMB	54.7	57.1	59.5	61.9	64.9	67.8	71.1	72.6	75.9
DeepWalk	56.0	63.0	70.2	75.5	80.1	85.2	85.3	87.8	90.3
LINE	55.0	58.6	66.4	73.0	77.6	82.8	85.6	88.4	89.3
node2vec	55.9	62.4	66.1	75.0	78.7	81.6	85.9	87.3	88.2
Naive Combination	72.7	82.0	84.9	87.0	88.7	91.9	92.4	93.9	94.0
TADW	86.6	88.2	90.2	90.8	90.0	93.0	91.0	93.4	92.7
CENE	72.1	86.5	84.6	88.1	89.4	89.2	93.9	95.0	95.9
CANE (text only)	78.0	80.5	83.9	86.3	89.3	91.4	91.8	91.4	93.3
CANE (w/o attention)	85.8	90.5	91.6	93.2	93.9	94.6	95.4	95.1	95.5
CANE	**86.8**	**91.5**	**92.2**	**93.9**	**94.6**	**94.9**	**95.6**	**96.6**	**97.7**

5.4.4　链接预测

如表 5.2、 5.3和 5.4所示，我们比较不同模型在保留不同比例的边的情况下（15% 到 95%）链接预测结果。需要注意的是，当我们仅仅保留 5% 的边时，网络中的许多节点变成孤立节点，使得网络表示学习模型不能学习出有效的节点表示。因此，我们忽略了不同模型在该训练比例下的结果。从这些结果中，我们可以发现：

表 5.3　HepTh 数据集链接预测结果 ($\alpha = 0.7, \beta = 0.2, \gamma = 0.2$)。

% 保留的边比例	15%	25%	35%	45%	55%	65%	75%	85%	95%
MMB	54.6	57.9	57.3	61.6	66.2	68.4	73.6	76.0	80.3
DeepWalk	55.2	66.0	70.0	75.7	81.3	83.3	87.6	88.9	88.0
LINE	53.7	60.4	66.5	73.9	78.5	83.8	87.5	87.7	87.6
node2vec	57.1	63.6	69.9	76.2	84.3	87.3	88.4	89.2	89.2
Naive Combination	78.7	82.1	84.7	88.7	88.7	91.8	92.1	92.0	92.7
TADW	87.0	89.5	91.8	90.8	91.1	92.6	93.5	91.9	91.7
CENE	86.2	84.6	89.8	91.2	92.3	91.8	93.2	92.9	93.2
CANE (text only)	83.8	85.2	87.3	88.9	91.1	91.2	91.8	93.1	93.5
CANE (w/o attention)	84.5	89.3	89.2	91.6	91.1	91.8	92.3	92.5	93.6
CANE	**90.0**	**91.2**	**92.0**	**93.0**	**94.2**	**94.6**	**95.4**	**95.7**	**96.3**

表 5.4　Zhihu 数据集链接预测结果 ($\alpha = 1.0, \beta = 0.3, \gamma = 0.3$)。

% 保留的边比例	15%	25%	35%	45%	55%	65%	75%	85%	95%
MMB	51.0	51.5	53.7	58.6	61.6	66.1	68.8	68.9	72.4
DeepWalk	56.6	58.1	60.1	60.0	61.8	61.9	63.3	63.7	67.8
LINE	52.3	55.9	59.9	60.9	64.3	66.0	67.7	69.3	71.1
node2vec	54.2	57.1	57.3	58.3	58.7	62.5	66.2	67.6	68.5
Naive Combination	55.1	56.7	58.9	62.6	64.4	68.7	68.9	69.0	71.5
TADW	52.3	54.2	55.6	57.3	60.8	62.4	65.2	63.8	69.0
CENE	56.2	57.4	60.3	63.0	66.3	66.0	70.2	69.8	73.8
CANE (text only)	55.6	56.9	57.3	61.6	63.6	67.0	68.5	70.4	73.5
CANE (w/o attention)	56.7	59.1	60.9	64.0	66.1	68.9	69.8	71.0	74.3
CANE	**56.8**	**59.3**	**62.9**	**64.5**	**68.9**	**70.4**	**71.4**	**73.6**	**75.4**

- 在不同数据集和训练比例下，我们提出的 CANE 模型比所有的基准方法都获得了显著的提升。这些结果表明，CANE 能够有效的对节点之间的关系进行建模，从而在链接预测任务上获得了有效的提升。
- 需要注意的是，CENE 和 TADW 两个方法在不同的训练比例下效果非常不稳定。具体来说，CENE 在训练比例低的情况下效果差，因为它与 TADW 等模型相比，包含了更多的模型参数，这些参数在训练数据不足时不能得到足够的优化。与 CENE 不同，TADW 在低训练比例的情况下表现较好，这是因为基于 DeepWalk 的方法即使在有限的边的情况下，也能够通过随机游走的方法来充分地探究网络结构，解决训练数据有限的问题。然而，TADW 在高训练比例的情况下表现较差，这是由于模型过于简单，而且会受到词袋假设的

限制。相较于上述两种方法，我们提出的 CANE 在不同训练比例的情况下都有着稳定的提升。这反映了 CANE 的鲁棒性和适应性。

- 通过引入互相注意力机制，CANE 学习到的上下文相关节点表示比不考虑注意力机制的模型学习到的上下文无关网络表示获得了显著的提升。这也验证了我们通过注意力机制来学习上下文相关表示向量的有效性和合理性。此外，这也验证了我们的假设，一个节点在与不同邻居交互时应该展现不同的角色。
- 在 CANE 中，我们假设每个节点表示包含基于结构的和基于文本的表示两部分，通过设计不同的目标函数，我们保证了网络节点的结构表示和文本表示的一致性和区分性。与只考虑文本表示的 CANE 相比，引入结构表示和文本表示的 CANE 也获得了明显的提升。这些结果表明了 CANE 引入结构表示和文本表示的有效性和合理性。

总结来说，上述观察结果表明，CANE 能够学习高质量的上下文相关的网络节点表示，能够有效地对节点之间的关系进行准确的建模，进而有助于节点之间的链接预测任务。链接预测的实验结果也验证了 CANE 的有效性和鲁棒性。

5.4.5　节点分类

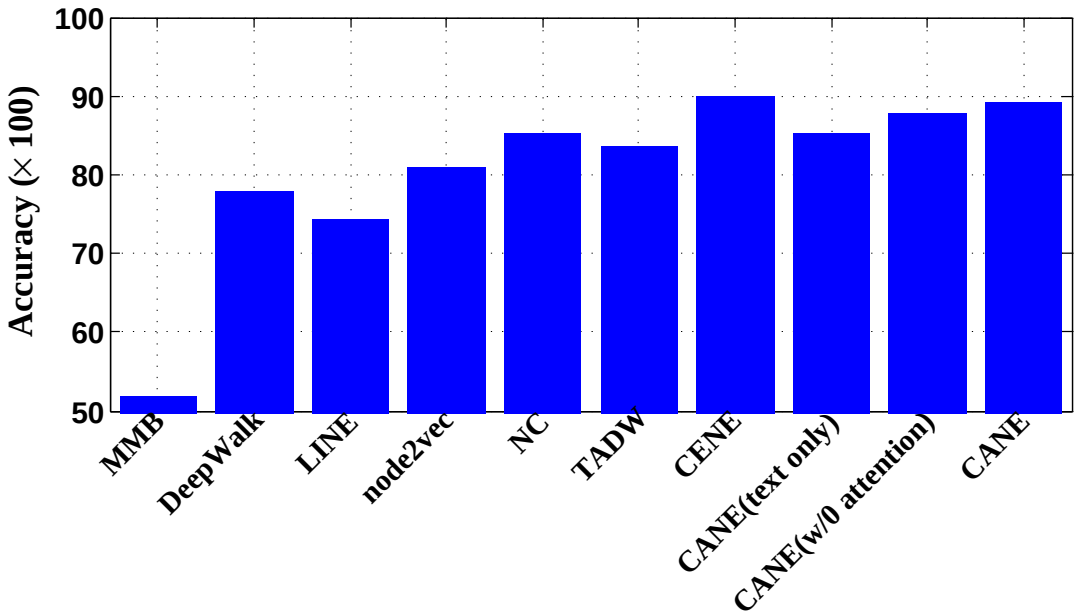


图 5.3　Cora 数据集节点分类结果。

在 CANE 中，我们能够根据节点交互的邻居不同来学习不同的节点表示。非常直观的是，这种上下文相关的节点表示能够自然的适用于链接预测任务。然而，

对于像节点分类、聚类等网络分析任务，我们需要一个全局的节点表示，而不是上下文相关的网络节点表示。

为了验证 CANE 也能够适用于其它的网络分析任务，我们通过一种简单的方式，来将节点的上下文相关的网络节点表示，转化为一个全局的上下文无关的网络节点表示。具体来说，我们对于一个节点所有的上下文相关的网络表示进行平均，来得到它的全局表示，如下所示：

$$\mathbf{u} = \frac{1}{N} \sum_{(u,v)|(v,u)\in E} \mathbf{u}_{(v)},$$

这里，N 表示节点 u 的上下文相关节点表示的数量。

利用上述方法生成的全局节点表示，我们进行了 2 层交叉验证，来汇报平均的节点分类准确率。如图 5.3所示，我们可以发现：

- CANE 在节点分类任务上获得了与最好的方法 CENE 可比的效果。这表明 CANE 学习到的上下文相关网络节点表示能够通过简单的方式转化为高质量的上下文无关的网络节点表示，也就是一个节点的全局表示。这也证明了 CANE 能够适用于其它的网络分析任务。
- 通过引入互相注意力机制，CANE 在节点分类任务上也获得了显著的提升。这个结果与链接预测任务中的结果一致，也表明了在 CANE 中引入互相注意力机制的合理性和有效性。

5.4.6　示例

为了更直观的展示 CANE 中互相注意力机制对于选择有效文本特征的能力，我们对于不同的节点对的文本注意力结果进行了热度图的可视化。如图 5.4所示，每个词都有着不同程度的背景色，颜色越深，说明该词的权重越大。每个词的权重都是根据 CANE 中的注意力权重计算得到。具体来说，我们首先根据公式 (5-11) 计算出每个卷积窗口的注意力权重。然后，我们将卷积窗口的权重分配到每个词上。最后，我们将一个词的注意力权重相加，得到它最终的权重。

我们提出的互相注意力机制，使得节点之间的关系变得显式和可解释。我们从 Cora 数据集中选取三个存在引用关系的论文，记为 A、B 和 C。从图 5.4中，我们发现，尽管 A 和 B 以及 A 和 C 之间都存在引用关系，但是 B 和 C 引用 A 的原因不同，它们也关注了 A 的不同文本信息。第一条边上，A 中的权重主要被分配到 “reinforcement learning" 等词上；而对于第二条边，A 的权重主要被分配到 “machine learning’", “supervised learning algorithms" 以及 “complex stochastic models" 等词上。

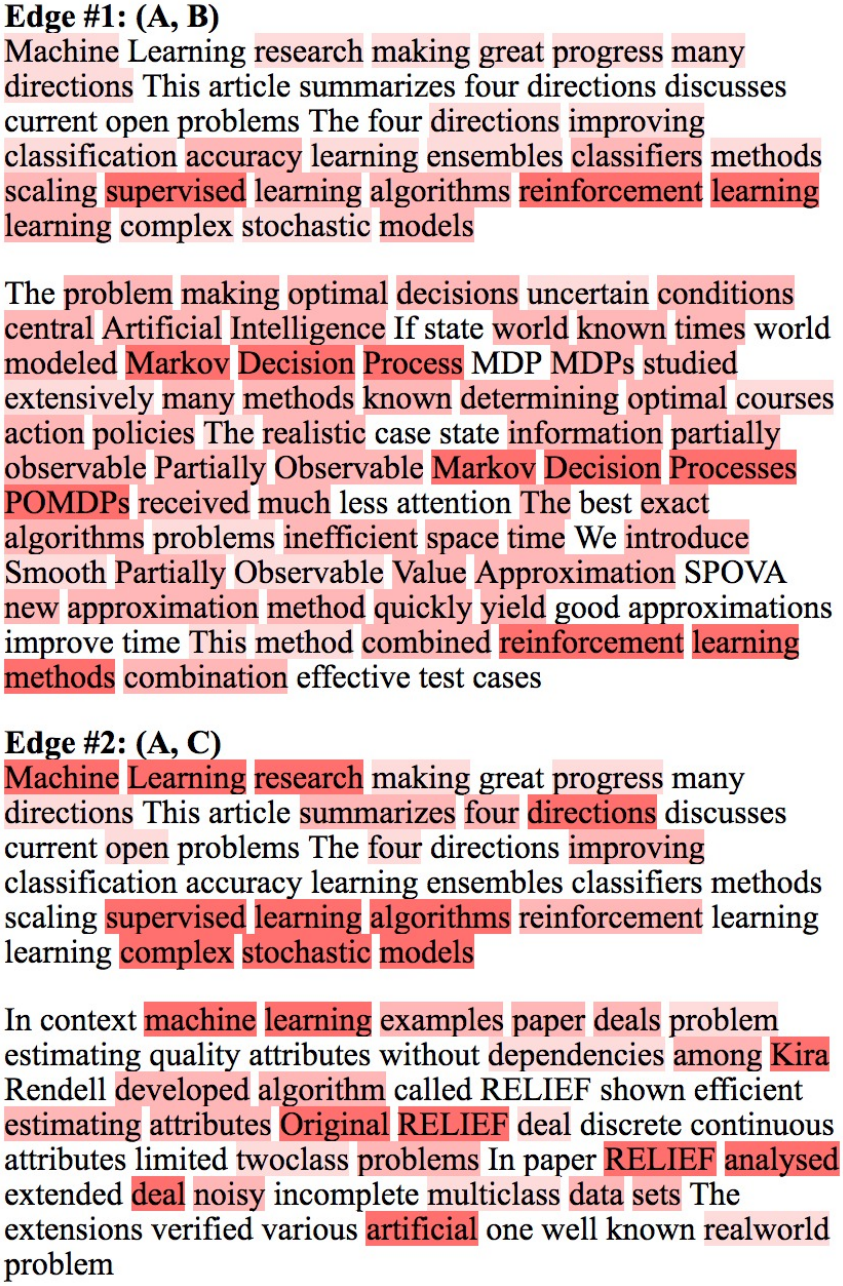


图 5.4　相互注意力机制可视化结果。

此外，A 中的关键词都能够在 B 和 C 中找到对应的词。因此，互相注意力机制通过赋予文本中不同的词不同的权重，来对论文之间的引用关系进行直观的解释。这里发现的节点对之间显著的关联关系也验证了互相注意力机制的有效性以及 CANE 对于节点之间关系进行准确建模的能力。

5.5　本章小结

在本章工作中，我们首次提出了上下文相关网络表示的概念，以及学习上下文相关网络表示的 CANE 模型。CANE 能够根据一个节点交互的邻居不同，来学习上下文相关的网络节点表示。具体来说，我们在基于文本的信息网络中引入互相注意力机制，来学习上下文相关的网络节点表示。在多个真实世界的网络数据集上的实验结果表明，CANE 能够显著的提升链接预测任务的效果。此外，CANE 学习到的上下文相关网络表示能够通过简单的方式组成高质量的上下文无关的网络表示。

第 6 章　面向社会关系抽取的隐式网络表示

在前一个章节中，我们介绍了在网络表示学习中考虑节点附加的文本信息，来对节点之间的交互关系进行建模，从而提高链接预测任务的效果。虽然这种网络表示学习方式能够对节点之间的关系进行一定程度的解释，但是这种解释方式不够直观。同时，已有的网络表示学习工作往往将网络中的边简化成二元或实数值，而忽略了边上丰富的语义信息。此外，已有的网络分析任务并不能很好的衡量网络表示学习模型对节点之间的显式关系进行建模的能力。

在本章工作[①]中，我们首先针对已有网络分析工作的不足，提出了社会关系抽取任务（Social Relation Extraction, SRE），来衡量网络表示学习模型对于节点之间显式的关系进行建模和预测的能力。进一步的，为了解决社会关系抽取问题，我们将节点之间的交互行为建模成平移机制，提出了基于平移的网络表示学习模型，TransNet。在真实的大规模社会网络数据集上的实验结果表明，我们提出的 TransNet 模型能够显著的提升社会关系抽取任务的效果。

6.1　问题描述

如何表示网络中的节点，对于节点分类[99]、聚类[100]、链接预测[101] 等网络分析任务至关重要。传统的基于符号的网络表示通常面临着稀疏性问题和计算效率问题。随着表示学习在图像、语音、自然语言处理等领域的发展，针对大规模社会网络的网络表示学习被提了出来。网络表示学习目的是为网络中节点学习一个低维实值的向量表示，这些向量表示往往会被当作特征向量，应用到进一步的网络分析任务中。

近些年来，许多网络表示学习的方法不断涌现。这些网络表示学习模型尝试根据节点的局部结构信息[1,6] 或者全局特征[9] 来学习有效的节点表示。此外，一些网络表示学习工作尝试将网络节点的异构信息融入节点的表示中，例如文本信息[10] 和标签信息[89,102]。

值得注意的是，大多数已有的网络表示学习工作忽略了边上丰富的语义信息。作为网络的重要组成部分，边通常会在网络表示学习模型和网络分析任务中，被简化成二元或者实数值。显然，这种简化不能够对边上丰富的语义信息进行很好

① 本章主要工作以 “TransNet: Translation-Based Network Representation Learning for Social Relation Extraction’’ 为题发表在 2017 年的国际学术会议 “The International Joint Conference on Artificial Intelligence (IJCAI’17)’’ 上。

的建模。在真实世界的网络中，节点之间的交互通常蕴含着丰富且多样的含义。例如，社交媒体中，对于同一个用户的关注行为可能是由于不同的原因；学术合作网络中，两个研究者与另外的研究者的合作行为也可能由于不同的研究兴趣。因此，将边上丰富的关系信息引入网络表示学习中非常必要。

图 6.1　社会关系抽取中关系的定义。

在本章的工作中，我们首先提出了社会关系抽取（Social Relation Extraction, SRE）的任务，来对社会网络中的关系进行建模和预测。社会关系抽取和知识图谱中的关系抽取非常相似。在知识图谱中的关系抽取中，最常采用的方法是基于平移的知识表示学习模型，例如 TransE[103]。与知识图谱中的关系抽取相比，在社会关系抽取中，没有预定义好的关系类别，同时，节点之间的关系通常隐藏在它们交互的文本信息中（例如，研究者之间合作发表的论文）。因此，如图 6.1所示，在社会关系抽取任务中，我们可以通过已有的自然语言处理技术，例如关键词抽取、命名实体识别等，从交互文本中抽取关键词或者命名实体，来表示节点之间的交互关系。

社会关系抽取任务不能很好的被已有的网络表示学习和知识表示学习方法解决。传统的网络表示学习模型在学习节点表示的过程中，往往会忽略边上丰富的语义信息；而典型的知识表示学习方法一般仅在边上标注了单独的标签时才会有着较好的表现。根据统计，在典型的知识图谱数据集 FB15k 中，只有 18% 的实体对之间存在着多关系标签，而社会网络数据集中的多标签的边的比例数倍于知识图谱中的比例。

为了解决该问题，我们提出了一个新颖的基于平移的网络表示学习模型，TransNet，来结合边上的多标签信息。受平移机制在词向量表示[5] 和知识表示[103],

中的成功应用的启发，我们将节点和边映射到统一的表示空间，并且采用平移机制来建模它们之间的交互，也就是说，一条边上尾节点的表示应该尽量接近头节点的表示加上边的表示。为了解决多标签的问题，在 TransNet 中，我们设计了一个自动编码器来根据标签集合学习边的表示。此外，自动编码器的解码部分能够用来预测未标注边的关系标签，也就是进行社会关系抽取。

为了评测社会关系抽取任务的效果，我们构建了三个真实的学术网络数据集。实验结果表明，TransNet 与传统的网络表示学习模型和 TransE 相比，在社会关系抽取任务上获得了显著的提升。这表明我们提出的 TransNet 模型对于关系建模和预测上的效果。

总结来说，本章工作主要有以下几点贡献：

- 我们首次提出了社会关系抽取任务，用来衡量网络表示学习模型对于节点之间关系的建模和预测能力。
- 我们提出了一个新颖的基于平移的网络表示学习模型 TransNet，来结合边上丰富的语义信息。TransNet 中的平移机制能够很好的对节点、关系表示之间的交互进行建模和预测。
- 我们收集并构建了三个不同规模的社会关系抽取数据集。实验结果表明，TransNet 在社会关系抽取任务上显著优于已有的网络表示学习方法和知识表示学习方法。

6.2　相关工作

近些年来，网络表示学习成为数据挖掘领域一个热门的研究方向。不同的网络表示学习方法不断被提出，这些方法可以大致被分为三个类别。DeepWalk[1], LINE[6], node2vec[7] 和 SDNE[8] 尝试根据节点的局部网络结构信息来学习网络节点表示。此外，一些工作尝试利用网络中存在的全局信息来提高网络节点表示质量，例如，GraRep[9] 和 MNMF[104]。此外，如何在网络表示学习过程中融合异构信息也十分重要。TADW[10] 通过矩阵分解来结合节点的文本信息；MMDW[89] 和 DDRW[102] 通过结合节点的类别标签信息，来学习有区分性的网络节点表示。

然而，这些已有的工作都没有充分利用边上丰富的语义信息，也不能对边上的关系进行有效的预测。值得注意的是，关系抽取已经成为知识图谱领域重要的任务[105–109]，目的是从已有的知识图谱或者文本中抽取关系事实来丰富知识图谱。在知识图谱领域，关系抽取通常被建模成关系分类任务，目前存在的许多大规模的知识图谱，例如，Freebase[110]、DBpedia[111] 以及 YAGO[112] 等，为训练关系分类器提供了有效的训练数据。然而，在社会网络中，通常没有显式的关系标注数据，

而人工进行关系的定义和标注费时费力。为了解决该问题，我们提出利用已有的 NLP 技术从节点交互的文本中自动构建显式的关系数据。

如何建模节点和边之间的关系对于关系的预测非常重要。在词向量表示学习领域，Mikolov et al.[5] 发现了词向量之间的平移现象，例如，“*King*"-“*man*"=“*Queen*"-“*Woman*"。在知识图谱领域， Bordes et al.[103] 将实体、关系看作在统一表示空间的平移操作，也就是 “*head*"+“*relation*"=“*tail*"。受这些平移机制的启发，我们假设在社会网络中网络节点之间同样存在着平移机制，并且提出了 TransNet 模型来对这种平移机制进行建模。

社会关系抽取与知识图谱中的关系抽取非常相关。知识图谱中的关系抽取是对已有的知识库进行扩展的重要技术。知识表示学习方法，例如 TransE[103]，已经成为知识图谱关系抽取的重要手段。

在本章工作中，我们提出了社会关系抽取任务，来对社会网络节点之间的关系进行建模和预测。与知识图谱中的关系抽取相比，社会关系抽取主要有两个显著的不同：

- 在知识图谱中，关系类别通常被很好的预定义。实体之间的关系也经过大量精确的人工标注。相反的是，社会关系抽取处理的是一个全新的场景，其中，节点之间的关系是隐式的，通常隐含在它们交互的文本信息中。
- 在社会网络中，节点之间的关系是动态变化的而且非常复杂，不能很好的用一个单独的标签表示。因此，通过交互文本中的关键词等集合来表示节点之间的关系非常直观有效。这些关键词能够很好的捕捉到节点之间复杂的语义信息，也能够使得节点之间的关系显式且可解释。社会关系抽取中对于关系的定义如图 6.1所示。

形式上，我们如下定义社会关系抽取任务。假设存在一个社会网络 $G = (V, E)$，其中 V 表示节点集合，$E \subseteq (V \times V)$ 为节点之间边的集合。此外，这些边部分被标注，标注的边的集合记为 E_L。不失一般性，我们将节点之间的关系定义为一个标签集合，而不是单独一个标签。具体来说，对于每条边 $e \in E_L$，对应的标签集合为 $l = \{t_1, t_2, \ldots\}$，其中每个标签 $t \in l$ 来自一个固定的标签词表 T。

如图 6.2所示，给定整个网络结构以及标注的标签集合 E_L，社会关系抽取目的是预测未标注边集合 E_U 中每条边对应的标签集合，也就是每条边对应的具体关系。这里，$E_U = E - E_L$ 表示未标注的边集合。

6.3　模型框架

在本章工作中，我们关注如何将边上的标签信息融合到网络表示学习模型中。

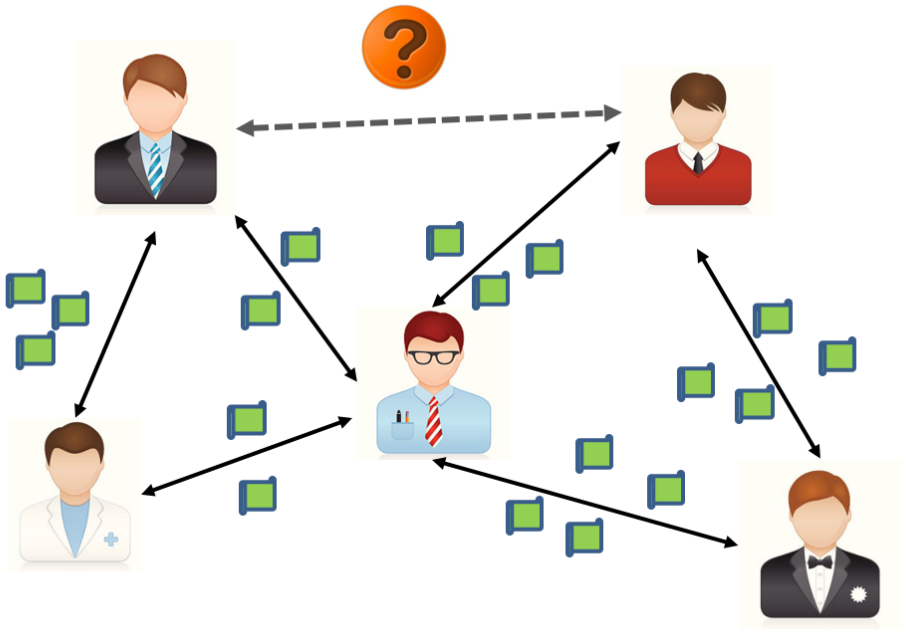


图 6.2　社会关系抽取示意图。

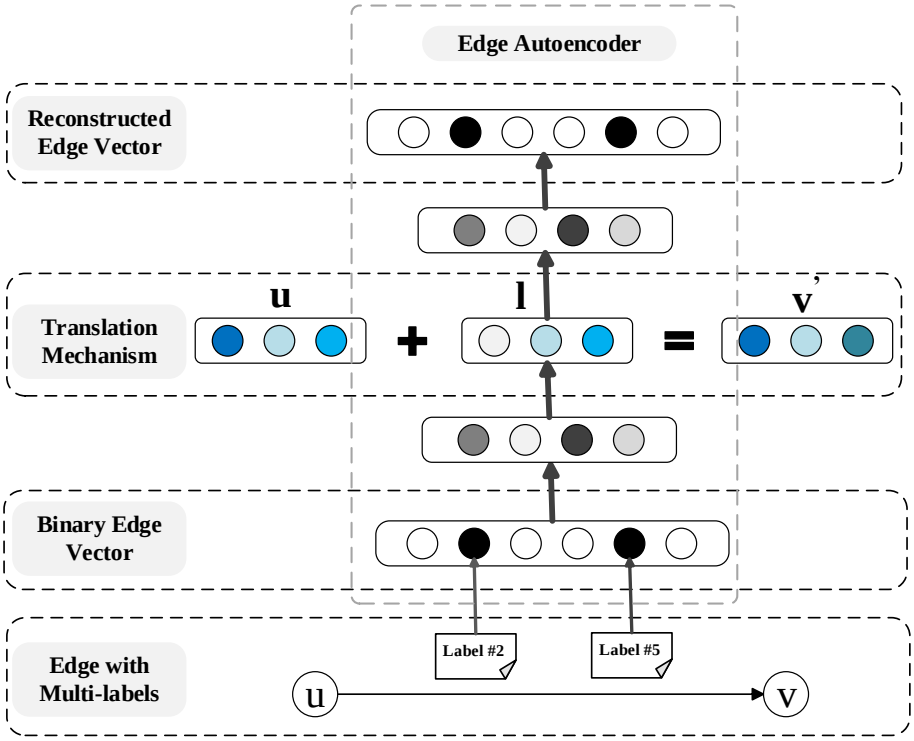


图 6.3　TransNet 模型示意图。

如图 6.3所示，TransNet 包含两个重要部分，也就是平移机制和边表示构建。在接下来的小节中，我们首先给出 TransNet 中平移机制的详细介绍，随后，我们介绍如何根据边上的标签集合构建边上关系的表示。最后，我们给出 TransNet 整

体的优化目标。

6.3.1　平移机制

受平移机制在词向量表示学习[5] 和知识表示学习[103] 的启发，我们假设社会网络中节点之间的交互也能够表示成表示空间中的平移操作。

具体来说，对于每条边 $e = (u, v)$ 和它对应的标签集合 l，节点 v 的表示向量要尽可能接近于节点 u 的表示向量加上边的表示向量。由于每个节点在 TransNet 中扮演两个角色，也就是头节点或者尾节点，我们对于每个节点 v 引入两个不同的表示 $\mathbf{v}$ 和 $\mathbf{v}'$ 来对应它的头节点表示和尾节点表示。随后，节点 u, v 和边 e 之间的平移机制可以形式化为：

$$\mathbf{u} + \mathbf{l} \approx \mathbf{v}'. \tag{6-1}$$

这里，$\mathbf{l}$ 表示由标签集合 l 得到的边的表示向量。具体计算细节将会在小节 6.3.2中详细介绍。

我们采用距离函数 $d(\mathbf{u} + \mathbf{l}, \mathbf{v}')$ 来衡量三元组 (u, v, l) 符合公式 (6-1) 的程度。实际应用中，我们采用 L_1 距离来作为距离函数。

根据上述定义，对于每个三元组 (u, v, l) 和它对应的负例 $(\hat{u}, \hat{v}, \hat{l})$，TransNet 的平移部分的损失函数如下：

$$\mathcal{L}_{trans} = \max(\gamma + d(\mathbf{u} + \mathbf{l}, \mathbf{v}') - d(\hat{\mathbf{u}} + \hat{\mathbf{l}}, \hat{\mathbf{v}}'), 0), \tag{6-2}$$

其中，$\gamma > 0$ 为间隔的超参数，$(\hat{u}, \hat{v}, \hat{l})$ 是从负例集合 N_e 中采样出的负例。负例集合的定义如下：

$$N_e = \{(\hat{u}, v, l) | (\hat{u}, v) \notin E\} \cup \{(u, \hat{v}, l) | (u, \hat{v}) \notin E\} \cup \{(u, v, \hat{l}) | \hat{l} \cap l = \emptyset\}. \tag{6-3}$$

在公式 (6-3) 中，头节点或尾节点被随机替换成另外的不相连的节点，标签集合被随机替换成不相交的标签集合。

公式 (6-2) 中的节点表示是模型的参数，而边的表示则是由对应的标签集合生成。具体构建过程会在下一小节详细介绍。

6.3.2　边表示构建

如图 6.3所示，我们采用了一个深层自动编码器来构建边的表示。自动编码器的编码部分包含多个非线性变换层，来将输入的标签集合映射到低维的表示空间。解码器部分的重构过程保证了边表示向量蕴含了输入的标签集合的全部信息。在接下来的部分，我们会详细介绍边表示构建的实现过程。

输入映射：我们首先将输入的标签集合映射成向量形式。具体来说，给定一个标签集合 $l = \{t_1, t_2, \ldots\}$，我们计算其对应的二元表示向量 $\mathbf{s} = \{\mathbf{s}_i\}_{i=1}^{|T|}$，其中，$\mathbf{s}_i = 1$ 如果 $t_i \in l$，否则 $\mathbf{s}_i = 0$。

非线性变换：将标签集合映射得到的二元表示向量 $\mathbf{s}$ 作为输入，自动编码器的编码和解码部分分别包含多层的非线性变换，如下所示：

$$\begin{aligned} \mathbf{h}^{(1)} &= f(\mathbf{W}^{(1)}\mathbf{s} + \mathbf{b}^{(1)}), \\ \mathbf{h}^{(i)} &= f(\mathbf{W}^{(i)}\mathbf{h}^{(i-1)} + \mathbf{b}^{(i)}), i = 2, \ldots, K. \end{aligned} \tag{6-4}$$

其中, K 表示非线性变换的层数，f 表示激活函数。$\mathbf{h}^{(i)}$, $\mathbf{W}^{(i)}$ 和 $\mathbf{b}^{(i)}$ 分别表示第 i 层的隐向量、变换矩阵和偏置向量。

具体来说，由于节点的表示向量为实值向量，因此我们采用 *tanh* 激活函数来得到中间的边表示向量 $\mathbf{l} = \mathbf{h}^{(K/2)}$。此外，由于输入向量 $\mathbf{s}$ 为二元向量，因此我们在最后一层采用 *sigmoid* 激活函数来得到重构的向量输出 $\hat{\mathbf{s}}$。

重构损失：自动编码器的作用是最小化输入向量和重构输出之间的距离。因此，自动编码器的损失函数可以形式化为：

$$\mathcal{L}_{rec} = ||\mathbf{s} - \hat{\mathbf{s}}||. \tag{6-5}$$

其中，为了与公式 (6-2) 保持一致，我们同样采用 L_1 距离来衡量输入向量和重构的输出向量之间的距离。

然而，由于输入向量的稀疏性，向量 $\mathbf{s}$ 中零元的数量远多于非零元的数量。这意味着自动编码器会趋向于重构出零元，而不是非零元。这种结果与我们的目标相违背。因此，我们对于不同的元素设置不同的权重，来重新定义公式 (6-5) 中损失函数，如下所示：

$$\mathcal{L}_{ae} = ||(\mathbf{s} - \hat{\mathbf{s}}) \odot \mathbf{x}||, \tag{6-6}$$

其中，$\mathbf{x}$ 是一个权重向量，$\odot$ 表示 Hadamard 乘积。对于 $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^{|T|}$，当 $\mathbf{s}_i = 0$ 时，

$\mathbf{x}_i = 1$，除此之外，$\mathbf{x}_i = \beta > 1$。这里 β 表示非零元的权重系数。

通过深层自动编码器，边的表示向量不仅包含对应标签集合的信息，也具备了预测节点之间关系（标签集合）的能力。

6.3.3　目标函数

为了确保节点表示和边表示之间的平移机制，以及边表示的重构能力，我们结合公式 (6-2) 和公式 (6-6) 中的损失函数，作为 TransNet 模型统一的目标函数。对于每个三元组 (u, v, l) 和对应的负例 $(\hat{u}, \hat{v}, \hat{l})$，TransNet 联合优化如下的损失函数

$$\mathcal{L} = \mathcal{L}_{trans} + \alpha[\mathcal{L}_{ae}(l) + \mathcal{L}_{ae}(\hat{l})] + \eta\mathcal{L}_{reg}. \tag{6-7}$$

这里，我们引入两个超参数 α 和 η 来平衡不同部分的权重。此外，$\mathcal{L}_{reg}$ 为一个 L_2 正则项，来防止过拟合，定义如下所示：

$$\mathcal{L}_{reg} = \sum_{i=1}^{K}(||W^{(i)}||_2^2 + ||b^{(i)}||_2^2). \tag{6-8}$$

为了防止过拟合，我们进一步采用 dropout[113] 来生成边的表示。最后，我们采用 Adam 优化算法[95] 来优化公式 (6-7) 中损失函数。

6.3.4　预测

利用学习到的网络节点表示和深层自动编码器，TransNet 能够预测 E_U 中未标注边上的标签集合信息，也就是进行社会关系抽取。

具体来说，给定未标注的边 $(u, v) \in E_U$，TransNet 假设节点 u 和 v 的表示与潜在的边表示符合公式 (6-1) 的平移机制。因此，我们能够通过 $\mathbf{l} = \mathbf{v}' - \mathbf{u}$ 得到近似的边的表示。接下来，我们对于得到的边的表示进行解码，来得到预测的标签向量 $\hat{\mathbf{s}}$。标签 t_i 对应的权重 $\hat{\mathbf{s}}_i$ 越大，意味着该标签越有可能属于标签集合 l。

6.4　实验结果

为了验证 TransNet 模型对于节点之间关系建模的有效性，我们在社会关系抽取任务上，与不同的基准方法在自动构建的社会网络数据集上进行了对比。

6.4.1　数据集

ArnetMiner[①] [114] 是一个为研究者提供检索服务的在线学术网络。该网络开放了大规模的学术合作网络数据集[②]，其中包含 1,712,433 名作者，2,092,356 篇论文以及 4,258,615 个合作关系。

在该网络中，研究者与不同的人在不同的主题上进行合作，而他们合作发表的论文能够反映出他们之间详细的合作关系。因此，我们通过如下步骤构建标注了关系的数据集：

- 我们从研究者的个人信息中收集了代表研究兴趣的词和短语，利用这些词项构建了标签词表。
- 对于每个合作关系，我们从他们合作发表论文的摘要中，过滤出标签词表中包含的关键词，把这些关键词当作对他们之间合作关系的标注。需要注意的是，由于合作网络中的边是无向的，我们把每条边用两条方向相反的有向边来替代。

此外，为了更好的探究不同模型的特点，我们构建了三个不同规模的数据集，包括 **Arnet-S**, **Arnet-M** 以及 **Arnet-L**。具体的数据集统计信息见表6.1。

表 6.1　数据集统计信息。

数据集	**Arnet-S**	**Arnet-M**	**Arnet-L**
节点	187,939	268,037	945,589
边	1,619,278	2,747,386	5,056,050
训练集	1,579,278	2,147,386	3,856,050
测试集	20,000	300,000	600,000
验证集	20,000	300,000	600,000
标签数量	100	500	500
多标签边的比例 (%)	42.46	63.74	61.68

6.4.2　基准方法

我们采用了如下网络表示学习方法进行对比，包括：

- **DeepWalk**[1] 在网络上进行随机游走，来生成节点的随机游走序列。然后，它把节点当作词，节点序列当作句子，采用训练词向量的 Skip-Gram[5] 模型，来学习网络节点表示。

① https://cn.aminer.org/

② https://cn.aminer.org/aminernetwork

- **LINE**[6] 定义了网络节点一阶邻近度和二阶邻近度，通过优化节点之间的联合概率和条件概率，来学习大规模网络的节点表示。
- **node2vec**[7] 通过扩展 DeepWalk 中的随机游走策略，利用 BFS 和 DFS 来更加有效的探究不同层次的网络结构。

对于这些网络表示学习模型，我们将社会关系抽取任务看作多标签分类任务。我们把一条边上的两个节点表示进行拼接，作为特征向量，来训练 one-vs-rest 逻辑回归[115] 多标签分类器。

此外，我们也与经典的知识表示学习模型 **TransE**[103] 进行对比。对于每个三元组 (u, v, l)，其中 $l = \{t_1, t_2, \ldots\}$，我们将其拆分成数个单标签的三元组 (u, v, t_i)，来作为 TransE 模型的训练数据。我们基于相似度的预测方法来预测边上的标签[103]。

6.4.3　评测指标和实验设置

为了进行公平的比较，我们和 TransE 一样，对于每个三元组 (u, v, t_i) 进行评测，其中 $t_i \in l$。此外，我们采用 *hits@k* 以及 *MeanRank*[103] 作为评测指标，分别记作 $h@k$ 和 MR。这里，*MeanRank* 表示标注标签在预测结果中的平均排序。*hits@k* 表示标注标签在预测结果的前 k 个的比例。需要注意的是，上述评测方法会低估把其它正确标签排在前面的模型的效果。因此，我们可以在排序之前把其它正确的标签过滤掉。我们将前面的评测设置记为 ‘Raw"，后面的记为 “Filtered"。

对于所有模型，我们设置表示向量维度为 100。对于 TransNet，正则项系数 η 为 0.001，Adam 优化算法对应的学习率为 0.001，间隔超参数 γ 取值为 1。此外，我们对于所有数据集采用了双层自动编码器，根据验证集的结果选取表现最好的取值 α 和 β。

表 6.2　Arnet-S 社会关系抽取结果 ($h@k$ ×100，$\alpha = 0.5$，$\beta = 20$)。

评测设置	Raw				Filtered			
指标	$h@1$	$h@5$	$h@10$	MR	$h@1$	$h@5$	$h@10$	MR
DeepWalk	13.88	36.80	50.57	19.69	18.78	39.62	52.55	18.76
LINE	11.30	31.70	44.51	23.49	15.33	33.96	46.04	22.54
node2vec	13.63	36.60	50.27	19.87	18.38	39.41	52.22	18.92
TransE	39.16	78.48	88.54	5.39	57.48	84.06	90.60	4.44
TransNet	**47.67**	**86.54**	**92.27**	**5.04**	**77.22**	**90.46**	**93.41**	**4.09**

表 6.3　Arnet-M 社会关系抽取结果 ($h@k$ ×100，$\alpha = 0.5$，$\beta = 50$)。

评测设置	Raw				Filtered			
指标	$h@1$	$h@5$	$h@10$	MR	$h@1$	$h@5$	$h@10$	MR
DeepWalk	7.27	21.05	29.49	81.33	11.27	23.27	31.21	78.96
LINE	5.67	17.10	24.72	94.80	8.75	18.98	26.14	92.43
node2vec	7.29	21.12	29.63	80.80	11.34	23.44	31.29	78.43
TransE	19.14	49.16	62.45	25.52	31.55	55.87	66.83	23.15
TransNet	**27.90**	**66.30**	**76.37**	**25.18**	**58.99**	**74.64**	**79.84**	**22.81**

表 6.4　Arnet-L 社会关系抽取结果 ($h@k$ ×100，$\alpha = 0.5$，$\beta = 50$)。

评测设置	Raw				Filtered			
指标	$h@1$	$h@5$	$h@10$	MR	$h@1$	$h@5$	$h@10$	MR
DeepWalk	5.41	16.17	23.33	102.83	7.59	17.71	24.58	100.82
LINE	4.28	13.44	19.85	114.95	6.00	14.60	20.86	112.93
node2vec	5.39	16.23	23.47	102.01	7.53	17.76	24.71	100.00
TransE	15.38	41.87	55.54	32.65	23.24	47.07	59.33	30.64
TransNet	**28.85**	**66.15**	**75.55**	**29.60**	**56.82**	**73.42**	**78.60**	**27.40**

6.4.4　实验结果和分析

表格 6.2、6.3和 6.4展示了不同数据集下的社会关系抽取结果。从这些表格中，我们有如下观察：

- 我们提出的模型 TransNet 与所有基准方法相比，在所有数据集上取得了显著且一致的效果提升。具体来说，TransNet 取得了比表现最好的基准方法 TransE10% 到 20% 左右的绝对提升。这些结果表明 TransNet 对于节点之间关系建模和预测的有效性和鲁棒性。
- 所有以往的网络表示学习方法在社会关系抽取任务上都表现很差。这是由于以往的网络表示学习方法在学习节点表示的过程中，忽略了边上丰富的语义信息。与之形成对比的是，TransE 和 TransNet 将边上的语义信息融合到节点表示中，因此获得了显著的提升。这表明了在网络表示学习中考虑边上的语义信息的重要性，以及平移机制对于节点之间关系进行建模的合理性。
- 与 TransNet 相比，TransE 表现较差，是因为该模型每次只能考虑一个边上的标签信息，这样会使得同一边上的不同标签表示趋向于相等。这种方式一定程度上符合知识图谱的场景，其中只有 18% 的实体对拥有多关系标签。然而对于社会关系抽取的场景，数据集的多标签的比例远大于知识图谱中的比例 (Arnet-S: 42%, Arnet-M: 64% Arnet-L: 62%)。因此，TransNet 能够同时考虑一条边上的多标签信息，结果表明 TransNet 能够很好的处理社会关系抽取中的

多标签问题。

- TransNet 在不同规模的网络中表现非常稳定。当标签数量变多时，TransNet 的效果仅有少量的下降（对于 filtered $h@10$，从 90% 到 80%)。与之形成对比的是，TransE 和其它网络表示学习方法的下降高达 20%。这个结果也表明了 TransNet 的可靠性和稳定性。

6.4.5　标签对比

表 6.5　Arnet-S 上的标签对比 (×100 for $h@k$)。

标签	最高频的 5 个标签				最低频的 5 个标签			
指标	$h@1$	$h@5$	$h@10$	MR	$h@1$	$h@5$	$h@10$	MR
TransE	58.82	85.68	91.61	**3.70**	52.21	82.03	87.75	5.65
TransNet	**77.26**	**90.35**	**93.53**	3.89	**78.27**	**90.44**	**93.30**	**4.18**

为了验证 TransNet 对于标签之间关系建模的优势，我们对比了 TransNet 和 TransE 在高频标签和低频标签上的表现。如表 6.5所示，我们展示 Arnet-S 数据集上 filtered *hits@k* 以及 *MeanRank* 的结果。

从该表中，我们发现，由于高频标签拥有充足的训练样例，TransE 在高频标签上的效果要显著优于低频标签上的效果。与 TransE 相比，TransNet 在高频和低频标签上表现更加稳定。这是因为 TransNet 使用了自动编码器来构建边的表示，这种方式能够利用标签之间的关联关系。这种关系能够对低频标签提供额外的信息，因此有利于对于低频标签的建模和预测。

6.4.6　参数敏感性分析

在 TransNet 中，有两个关键的超参数 α 和 β。因此，我们在 Arnet-S 数据集上，探究这些参数的敏感性。

参数 β 用来平衡自动编码器中零元和非零元的权重。实际应用中，我们在利用公式 (6-6) 初始化自动编码器参数时，利用重构的效果来决定 β 的取值。在图 6.4的左侧，我们展示不同的 β 取值下，验证集在评测指标 *hits@1* 上的变化趋势。从该图中，我们发现，重构的效果在数轮后变得稳定。当 β 取值过小时，例如 $\beta = 5$，自动编码器会倾向于重构零元而不是非零元，这样会导致重构的效果非常差。

参数 α 控制平移机制和自动编码器两者损失函数的比重。当固定最优的 β 取值后，我们在图 6.4右侧展示 α 在不同数量级的取值下，验证集在评测指标 $h@10$ 上的变化情况。我们发现，TransNet 的效果会随着训练轮数的增加逐渐上升，随后

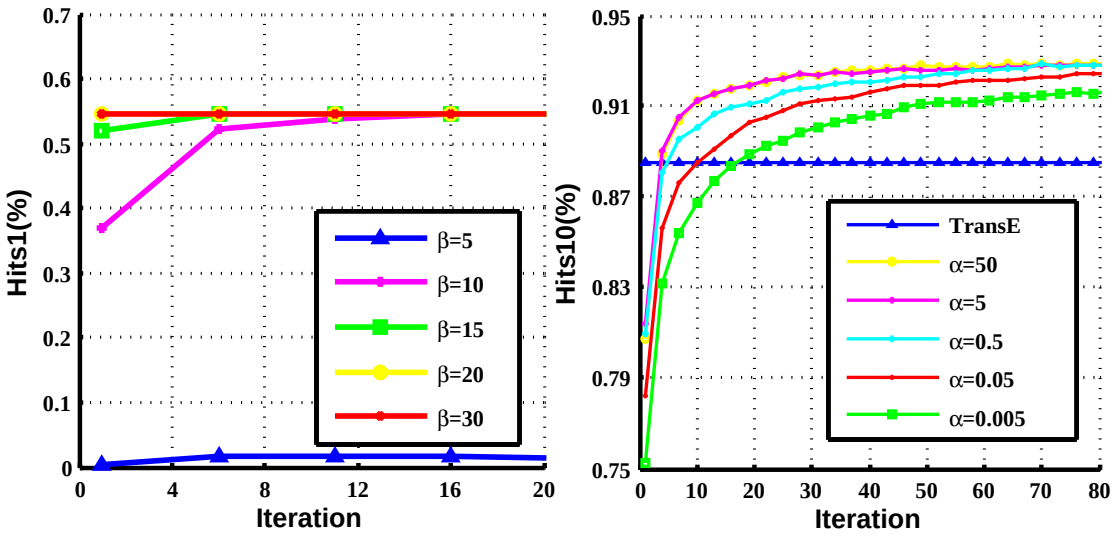


图 6.4　参数敏感性分析。

趋于稳定。当 α 取不同值时，所有的 TransNet 模型都能够在 20 轮以内超过 TransE 的效果，然后最终效果逐渐稳定。

上述两个敏感性分析结果表示，TransNet 在实际场景中容易训练，不容易过拟合，而且对超参数的设置不敏感。

6.4.7　示例

表 6.6　对于不同邻居推荐的 top-3 标签。

邻居节点	TransE	TransNet
Matthew Duggan	**ad hoc network**; wireless sensor network; wireless sensor networks	**management system**; **ad hoc network**; wireless sensor
K. Pelechrinis	**wireless network**; wireless networks; ad hoc network	**wireless network**; wireless sensor network; **routing protocol**
Oleg Korobkin	**wireless network**; wireless networks; **wireless communication**	**resource management**; **system design**; **wireless network**

我们从 Arnet-S 数据集中选取了一个例子来展示 TransNet 的有效性。被选取的研究者为 “A. Swami"，我们在表 6.6中展示了 TransE 和 TransNet 针对不同的合作者推荐的标签结果。在该表中，加粗的标签为推荐正确的标签。我们发现，TransE 和 TransNet 都能够根据不同邻居推荐合理的标签，能够反映出不同的合作主题。然而，对于一个特定的邻居，TransE 由于基于相似度的推荐方式限制，只能够推荐同

质化的标签。与之相比，TransNet 由于使用了自动编码器，推荐的标签更加多样，更有区分性。

6.5　本章小结

在本章工作中，我们提出了社会关系抽取的任务，来衡量网络表示学习模型对于关系建模和预测的能力。此外，我们提出了基于平移机制的网络表示学习模型 TransNet，来利用平移机制对节点之间的交互行为建模。针对社会关系抽取的实验结果表明，TransNet 能够有效的利用边上丰富的语义信息并将其编码到节点表示中，因此在社会关系抽取任务上取得了显著的效果提升。

第 7 章　社区优化隐式网络表示

在前面的章节中，我们介绍了在网络表示学习中融合节点的标签类别信息、文本信息以及边上的标签信息，来提高节点分类、链接预测、社会关系抽取等网络分析任务的效果。然而，这些方法都是基于网络节点局部的结构信息或者异构信息，没有考虑网络全局的结构特征。

在本章工作①中，我们针对典型的网络全局特征，也就是社区，提出了社区优化的网络表示学习模型（Community-enhanced NRL，CNRL）。具体来说，我们利用网络与文本之间的类比关系，将网络中的社区看作文本中的主题，在网络表示学习模型中结合主题模型，来同时训练网络节点的表示和进行社区发现。通过引入全局的社区信息，CNRL 在节点分类、链接预测、社区发现三个典型网络分析任务上都取得了显著的效果提升。

7.1　问题描述

随着大规模社交网络的发展，网络数据的规模不断增长，这些海量的网络数据为网络分析任务带来了巨大的挑战。为了对大规模社会网络进行高质量的特征表示，网络表示学习开始受到研究者的重视。网络表示学习目的是根据节点的网络结构信息，为节点学习一个低维实值的向量表示。

大多数已有的网络表示学习方法根据节点的局部结构信息来学习节点表示。例如，DeepWalk[1] 利用随机游走策略来探究节点周围邻居的信息，然后最大化由节点预测其随机游走序列中的局部邻居节点的概率。LINE[6] 对于节点对之间的一阶邻近度和二阶邻近度进行建模，然后最大化由当前节点预测邻居节点的概率。可以看到，DeepWalk 和 LINE 中的上下文节点均为局部的邻居节点。尽管 DeepWalk 和 LINE 的一些扩展方法，例如，node2vec[7] 和 GraRep[9]，能够探究更大范围的上下文信息，然而它们对于结构信息的考虑仍然受限，只能够反映出网络的局部特征。

如图 7.1所示，在一个典型的复杂网络中，节点通常会属于多个不同的社区。在社区内部，节点之间紧密相连[17]。同一社区里的节点通常会共享同样的属性。例如，有着相同教育背景的 Facebook 的用户，例如学校或者专业，往往倾向于形成一个社区[116]。因此，社区结构信息是网络节点一个非常重要的全局特征，有助于

① 本章主要工作以 “A Unified Framework for Community Detection and Network Representation Learning’’ 发表在 “IEEE Transactions on Knowledge and Data Engineering (TKDE)’’ 上。

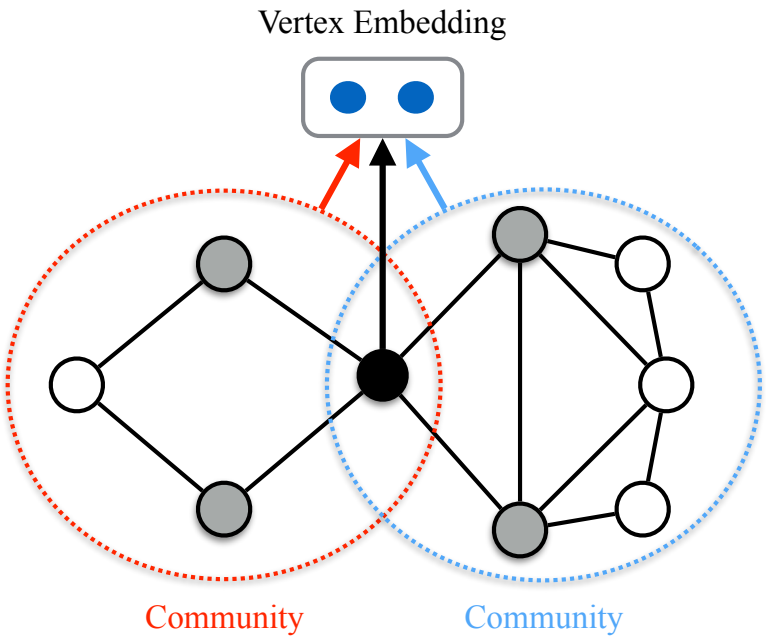


图 7.1　网络节点的社区特征：一个节点可能属于多个社区，该节点的表示向量应该受到它的邻居节点的影响以及所属社区的影响。

网络表示学习和相应的网络分析任务。受此启发，我们提出了一个统一的模型，社区优化的网络表示（Community-enhanced NRL, CNRL），来同时进行网络表示学习和社区发现。

需要指出的是，CNRL 受到 DeepWalk 对于文本和网络结构之间进行类比的启发。在 DeepWalk 中，作者发现，随机游走序列中的节点频度遵循着幂律分布，这与文本语料中词的分布类似。因此，通过把节点看作词，把节点序列看作句子，DeepWalk 直接采用词向量表示学习的模型来学习网络节点的表示。更进一步，如图 7.2所示，我们扩展了网络结构和文本之间的类比关系，假设在节点和随机游走序列之间存在中间的状态，也就是社区，这和词与文本之间存在主题相对应。这个假设非常直观，而且已经通过我们的实验和分析得到了验证。基于该假设，我们提出了 CNRL 模型，通过采用主题建模的方法来进行社区检测和学习网络表示。

图 7.3展示了 CNRL 的基本思想。在 CNRL 中，我们认为每个节点属于多个社区，这些社区之间相互重叠。和传统的基于局部信息的网络表示学习方法不同的是，CNRL 能够通过节点的局部上下文信息和全局的社区信息来学习节点表示。

在 CNRL 中，决定节点序列中每个节点的社区归属非常重要。我们假设节点对于社区的偏好对应着文本中词对于主题的偏好。因此，一个序列中的每个节点会根据节点的社区分布以及序列的社区分布，被分配一个特定的社区。随后，每个节点和它分配的社区被同时用来预测序列中的上下文节点。这样，我们能够学习到节点和社区的表示。在学习过程中，每个节点的社区分布也会迭代更新。

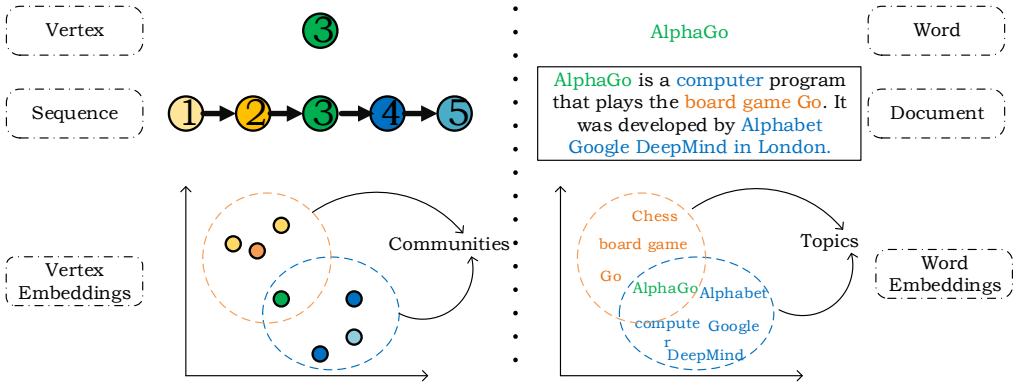


图 7.2　网络表示学习和文本建模之间的类比关系：网络中的节点、序列和社区分别对应文本中的词、文档和主题。

CNRL 通过引入社区信息来增强网络节点表示的质量，能够提高节点分类、链接预测等网络分析任务的效果。我们在两个典型的基于随机游走的网络表示学习模型，DeepWalk 和 node2vec 上，实现了 CNRL 模型。在多个真实的网络数据集上的结果表明，CNRL 能够显著的提高多个网络分析任务的效果。

总结来说，本章工作主要有以下几点贡献：

- 我们探究了网络中的社区与文本中主题的类比关系，将全局的社区信息引入到网络表示学习中，来提高网络节点表示的质量。我们提出的 CNRL 模型能够同时学习网络节点表示和进行社区发现。
- 在 CNRL 中，我们设计了两种有效的社区分配算法，包括基于统计的和基于表示的方法。这两种方法角度不同，但都表现优异。
- 为了对模型进行评测，我们在多个不同的数据集上进行了节点分类和链接预测实验。实验结果表明，CNRL 与其它模型相比获得了显著且一致的效果提升。
- 此外，我们提出的 CNRL 模型能够从不同的维度进行有重叠的社区检测。为了验证 CNRL 的有效性和可扩展性，我们可视化了 CNRL 检测出的不同维度的社区，并且与传统的社区发现方法进行了实验对比。

7.2　相关工作

7.2.1　社区检测

根据 Yang et al.[117] 的定义，网络中的一个社区是指一组节点的集合，社区中的节点互相之间紧密相连，而社区之间连接稀疏。在实际的网络中，同一个社区内部的节点往往共享相同的属性或者扮演同样的角色[118]。

从网络中进行社区检测一直是社会科学领域重要的研究课题。传统的方法一

般是将网络节点划分到不同的集合中，也就是不重叠的社区检测。已有的非重叠社区检测工作主要包括基于聚类的方法[119]，基于模块性的方法[17,118,120]，谱算法[121]，随机块方法 (stochastic block models, SBM)[122,123] 等等。这些方法主要的问题是不能够检测重叠的社区，因此与许多实际场景不相符。为了解决该问题，CPM[124] 通过对重叠的 k-cliques 进行合并来得到重叠的社区。 Ahn et al.[125] 通过边聚类的方法对边进行划分，来进行重叠的社区检测。

近些年来，社区关联算法能够有效的进行重叠社区发现[126–129]。社区关联算法一般会预定义好网络中社区的数量，然后为每一个节点学习一个社区的强度向量。例如， Yang et al.[129] 提出了非负矩阵分解的方法，来计算得到节点社区的关联矩阵。我们提出的模型遵循社区关联算法的设置，利用一个非负向量来表示每个节点对于社区的从属关系。

7.2.2　网络表示学习

如前文所述，大多数已有的网络表示学习模型仅仅考虑节点的局部结构信息，忽略了网络的全局特征。尽管基于随机游走的方法，例如 DeepWalk 和 node2vec，能够对相关邻居节点进行深层的探究，但是这种考虑上下文节点的方式会受到上下文窗口大小的限制[10,11]。此外，社区信息也有在这些方法中被显式的利用。 Wang et al.[104] 提出了一种基于非负矩阵分解的方法（modularized nonnegative matrix factorization，MNMF），来进行非重叠的社区检测和学习网络节点表示。然而，该方面有两个缺点。首先，MNMF 只能够检测非重叠的社区，也就是说，每个节点仅仅属于一个特定的社区，这种假设与实际场景存在偏差。此外，这种基于矩阵分解的模型优化复杂度为 $O(n^2m + n^2k)$ (n 为节点数量，m 为表示维度，k 为社区数量)。因此，MNMF 复杂度较高，不能处理大规模的网络。 Cavallari et al.[130] 提出了 ComE 模型，来同时学习节点表示和进行社区检测。在 ComE 中，每个社区被表示成一个高维高斯分布，来表示该社区中的网络节点的分布情况。通过精心设计的优化方法，ComE 能够在边的数量的线性时间内得到优化。然而，ComE 仅仅尝试求解最能够拟合节点表示的高维高斯分布，而没有显式的考虑社区的网络结构信息。因此，在本章工作中，我们探究文本中的主题和网络中的社区之间的类比关系，提出了基于社区优化的 CNRL 模型。CNRL 模型能够简单有效的结合到已有的网络表示学习模型中。

7.3　模型框架

7.3.1　问题定义

我们将一个网络记为 $G = (V, E)$，其中 V 表示节点集合，$E \subseteq (V \times V)$ 表示节点之间边的集合。$(v_i, v_j) \in E$ 表示节点 v_i 和 v_j 之间存在着边。对于每个节点 $v \in V$，网络表示学习的目的是为其学习一个向量表示 $\mathbf{v} \in \mathbb{R}^d$。这里，$d$ 为表示空间的维度。

网络 G 包含 K 个社区，记为 $C = \{c_1, \ldots, c_K\}$。不失一般性，我们假设这些社区是重叠的，也就是说，一个节点可能属于多个社区。这里，我们将节点 v 属于社区 c 的概率记作 $\Pr(c|v)$，社区 c 中包含该节点的概率记作 $\Pr(v|c)$。在 CNRL 中，我们也会学习每个社区 c 在同一表示空间的表示向量，记作 $\mathbf{c} \in \mathbb{R}^d$。

在接下来的小节中，我们首先介绍 DeepWalk 模型。随后，我们在 DeepWalk 模型的基础上进行扩展，实现 CNRL 模型。由于 node2vec 模型和 DeepWalk 模型仅在节点序列的生成方法上存在不同，不会影响 CNRL 的实现方法，因此，我们不再赘述基于 node2vec 的 CNRL 实现方法。

7.3.2　DeepWalk

DeepWalk[1] 首先会在网络 G 上进行随机游走，生成随机游走序列集合 $S = \{s_1, \ldots, s_N\}$。其中，每个随机游走序列可以表示为 $s = \{v_1, \ldots, v_{|s|}\}$。

随后，DeepWalk 将每个节点序列 s 看作词序列，把每个节点看成词，利用训练词向量的 Skip-Gram[5] 模型来根据序列集合 S 学习网络节点表示。

具体来说，给定节点序列 $s = \{v_1, \ldots, v_{|s|}\}$，每个节点 v_i 在滑动窗口中的局部上下文节点集合为 $\{v_{i-t}, \ldots, v_{i+t}\} \setminus \{v_i\}$。依照 Skip-Gram 模型，DeepWalk 通过最大化中心节点预测上下文节点的对数概率来学习节点表示，如下所示：

$$\mathcal{L}(s) = \frac{1}{|s|} \sum_{i=1}^{|s|} \sum_{i-t \le j \le i+t, j \ne i} \log \Pr(v_j|v_i), \tag{7-1}$$

其中，v_j 是节点 v_i 的上下文节点，概率 $\Pr(v_j|v_i)$ 通过 softmax 函数定义如下：

$$\Pr(v_j|v_i) = \frac{\exp(\mathbf{v}'_j \cdot \mathbf{v}_i)}{\sum_{v \in V} \exp(\mathbf{v}' \cdot \mathbf{v}_i)}. \tag{7-2}$$

这里，每个节点 v 包含两个表示向量，也就是作为中心节点的 $\mathbf{v}_i$ 和作为上下文节点的 $\mathbf{v}'$。

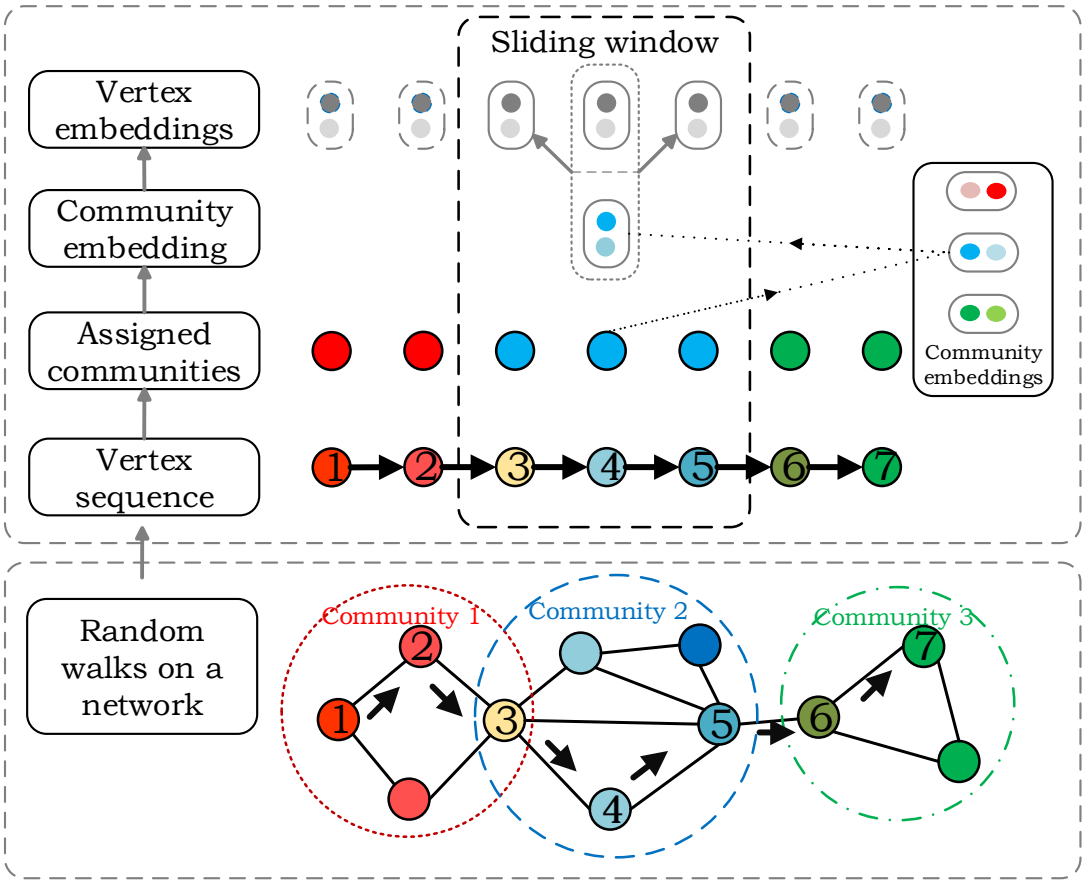


图 7.3　CNRL 模型示意图。

7.3.3　Community-enhanced DeepWalk

给定节点序列，DeepWalk 模型最大化上下文窗口中两个节点之间的局部概率。也就是说，节点的共现仅仅依赖于两个节点之间的吸引力，而忽略了全局的特征。在社交网络中，一个重要的全局特征是同质性（homophily），物以类聚，人以群分（“birds of a feather flock together”[18]）。这些分享相似属性的节点往往倾向于组成社区。

社区信息为网络节点提供了重要的上下文信息。为了在网络表示学习中考虑社区信息，来丰富节点的上下文信息，我们作出如下两个关于节点、序列和社区的假设：

假设 *1*： 网络中的每个节点属于多个不同的社区，且对于不同的社区有不同的偏好 $\Pr(c|v)$。同样的，每个随机游走序列也拥有对应的社区分布 $\Pr(c|s)$。

假设 *2*： 一个特定随机游走序列中的节点属于一个特定的社区，该社区由序列的社区分布 $\Pr(c|s)$ 和社区对于节点的分布 $\Pr(v|c)$ 共同决定。

基于上述假设，我们设计了迭代的方法来进行社区检测和学习网络节点、社区的表示。如图 7.3所示，这两个步骤包括，(1) **社区分配：** 我们为每个节点序列

中的节点分配一个离散的社区标签。（2）**表示学习**：给定一个节点和它对应的社区标签，我们尝试优化由该节点和社区标签预测上下文节点的概率，从而学习节点的表示和社区的表示。

7.3.3.1　社区分配

对于序列 s 中的一个节点 v，我们计算社区 c 的条件概率，如下所示：

$$\Pr(c|v,s) = \frac{\Pr(c,v,s)}{\Pr(v,s)} \propto \Pr(c,v,s). \tag{7-3}$$

根据我们的假设，联合概率 $\Pr(c,v,s)$ 可以形式化为：

$$\Pr(c,v,s) = \Pr(s)\Pr(c|s)\Pr(v|c), \tag{7-4}$$

其中，$\Pr(v|c)$ 表示节点 v 在社区 c 中的角色，$\Pr(c|s)$ 表示序列 s 和社区 c 的密切程度。根据公式 (7-3) 和公式 (7-4)，我们可以得到：

$$\Pr(c|v,s) \propto \Pr(v|c)\Pr(c|s). \tag{7-5}$$

在本章工作中，我们提出以下两种方式来计算 $\Pr(c|v,s)$：

基于统计的分配：依照主题模型 Latent Dirichlet Allocation (LDA)[66] 中吉布斯采样的参数估计方法，我们利用统计的方法计算条件概率 $\Pr(v|c)$ 和 $\Pr(c|s)$：

$$\Pr(v|c) = \frac{N(v,c)+\beta}{\sum_{\tilde{v}\in V} N(\tilde{v},c) + |V|\beta}, \tag{7-6}$$

$$\Pr(c|s) = \frac{N(c,s)+\alpha}{\sum_{\tilde{c}\in C} N(\tilde{c},s) + |K|\alpha}. \tag{7-7}$$

这里，α 和 β 是吉布斯采样中的平滑因子[66]。$N(v,c)$ 表示节点 v 被分配到社区标签 c 的频度，$N(c,s)$ 表示序列 s 中的节点被分配社区 c 的频度。$N(v,c)$ 和 $N(c,s)$ 都会随着社区分配的变化不断更新。

基于表示的分配：由于 CNRL 可以学习到节点和社区的表示向量，因此，我们可以根据这些表示向量来计算它们之间的条件概率。对于条件概率 $\Pr(c|s)$，我

们可以形式化为：

$$\Pr(c|s) = \frac{\exp(\mathbf{c} \cdot \mathbf{s})}{\sum_{\tilde{c} \in C} \exp(\tilde{\mathbf{c}} \cdot \mathbf{s})}, \tag{7-8}$$

其中，$\mathbf{c}$ 表示社区 c 对应的表示向量，$\mathbf{s}$ 为序列 s 的表示向量，可以通过序列中所有节点表示求平均得到。

实际上，我们也可以通过同样的方式来计算 $\Pr(v|c)$：

$$\Pr(v|c) = \frac{\exp(\mathbf{v} \cdot \mathbf{c})}{\sum_{\tilde{v} \in V} \exp(\tilde{\mathbf{v}} \cdot \mathbf{c})}. \tag{7-9}$$

然而，使用公式 (7-9) 会极大的影响模型的效果。因此，在实际应用中，我们只通过表示向量来计算 $\Pr(c|s)$，使用基于统计的方式来计算 $\Pr(v|c)$。

根据计算出的条件概率 $\Pr(v|c)$ 和 $\Pr(c|s)$，我们根据公式 (7-5) 为序列 s 中的每个节点 v 分配一个离散的社区标签 c。

7.3.3.2　表示学习

给定节点序列 $s = \{v_1, \ldots, v_{|s|}\}$，对于每个节点 v_i 和它分配的社区标签 c_i，我们通过最大化利用 v_i 和 c_i 预测上下文节点的概率来学习节点和社区的表示，目标函数如下所示：

$$\mathcal{L}(s) = \frac{1}{|s|} \sum_{i=1}^{|s|} \sum_{i-t \le j \le i+t, j \ne i} \log \Pr(v_j|v_i) + \log \Pr(v_j|c_i), \tag{7-10}$$

其中，$\Pr(v_j|v_i)$ 的计算方式和公式 (7-2) 一致。对于 $\Pr(v_j|c_i)$，我们同样采用 softmax 来计算：

$$\Pr(v_j|c_i) = \frac{\exp(\mathbf{v}'_j \cdot \mathbf{c}_i)}{\sum_{\tilde{v} \in V} \exp(\tilde{\mathbf{v}}' \cdot \mathbf{c}_i)}. \tag{7-11}$$

在实际场景中，我们不会显式的计算这些表示向量之间的条件概率，而是通过负采样算法[1] 来近似计算这些条件概率。

7.3.3.3　优化的节点表示

在完成上述表示学习过程之后，我们可以得到节点和社区的表示向量，以及节点的社区分布 $\Pr(c|v)$。根据这些结果，我们可以构建出优化的节点表示，记作

$\hat{\mathbf{v}}$。优化的节点表示包含了节点局部的网络结构信息和全局的社区信息。

具体来说，$\hat{\mathbf{v}}$ 包含两个部分，一部分是原始的节点表示 $\mathbf{v}$，一部分是它的社区表示 $\mathbf{v}_c$，这里的社区表示计算方法如下：

$$\mathbf{v}_c = \sum_{\tilde{c} \in C} \Pr(\tilde{c}|v)\tilde{\mathbf{c}}. \tag{7-12}$$

随后，我们将这两部分进行拼接，得到最终的优化的节点表示 $\hat{\mathbf{v}} = \mathbf{v} \oplus \mathbf{v}_c$。

CNRL 详细的训练过程见如算法 1所示。

Algorithm 1 CNRL 训练过程。

Require: graph $G = (V, E)$, community size K, window size t

Ensure: vertex embedding $\mathbf{v}$, context embedding $\mathbf{v}'$, community distribution $\Pr(v|c)$, community embedding $\mathbf{c}$

$S \leftarrow \text{SamplePath}(G)$
Initialize $\mathbf{v}$ and $\mathbf{v}'$ by Skip-Gram with S
Assign a community for each vertex in $\mathcal{S}$ randomly
for $iter = 1 : L$ **do**
    for each vertex v_i in each sequence $s \in S$ **do**
        Calculate Eqs. (7-6) and (7-7) w/o current assignment
        Assign a community c_i for v_i by Eq. (7-5)
    end for
end for
while not convergent **do**
    for each vertex v_i in each sequence $s \in S$ **do**
        Calculate Eq. (7-5) with Eq. (7-7) or Eq. (7-8)
        Assign a community c_i for v_i by Eq. (7-5)
        for each $j \in [i - t : i + t]$ **do**
            SGD on $\mathbf{v}$, $\mathbf{v}'$ and $\mathbf{c}$ by Eqs. (7-10) and (7-11)
        end for
    end for
end while

7.3.4　复杂度分析

CNRL 的训练过程包含两个部分，主题模型的吉布斯采样和表示学习。需要注意的是，基于表示的社区分配方法也需要预训练主题模型，来计算 $\Pr(v|c)$。基于吉布斯采样的 LDA 模型优化复杂度为 $O(LKnw\gamma)$，其中，L 表示循环次数，K 表示社区数量，n 表示节点数量，w 表示每个节点生成的随机游走序列数目，γ 为每个序列的长度。对于表示学习，基于 DeepWalk 的 CNRL 复杂度为 $O(n\log n)$，基于 node2vec 的模型复杂度为 $O(n\log n + na^2)$。这里，a 为节点平均度数。因此，CNRL 的训练复杂度为 $O(n(LKw\gamma + \log n + a^2))$。在实际训练过程中，CNRL 的两个模块都可以通过成熟的并行算法进行加速，例如，PLDA[131]，PLDA+[132] 以及 Skip-Gram[5]。

7.4　实验结果

在实验部分，我们探究了 CNRL 模型在节点分类、链接预测以及社区发现上的效果，并与不同的基准方法进行对比。

7.4.1　数据集

表 7.1　数据集统计信息。

数据集	Cora	Citeseer	Wiki	BlogCatalog
节点	2,708	3,312	2,405	10,312
边	5,429	4,732	15,985	333,983
类别	7	6	19	47
平均度数	4.01	2.86	6.65	32.39

如表 7.1所示，我们在四个广泛采用的网络数据集上进行了实验，包括：Cora、Citeseer、Wiki 以及 BlogCatalog。这些数据集属于不同的领域，而且拥有各自的特点，例如稀疏性、标签数量等等。

- **Cora.** Cora[①]是由 McCallum et al.[82] 构建的学术论文引用数据集。该数据集包含 2,708 篇机器学习相关论文，这些论文被划分为 7 个类别。论文之间的引用关系构成了网络。
- **Citeseer.** Citeseer 是由 McCallum et al.[82] 构建的另外一个学术论文数据集。该数据集包含 3,312 篇论文以及 4,732 条引用关系。这些论文被划分为 6 类。

① https://people.cs.umass.edu/ mccallum/data.html

- **Wiki.** Wiki[133] 是维基百科页面之间的链接网络。该数据集包含 2,405 个 web 页面，属于 19 个不同的类别。它们之间存在 15,985 条链接关系。该数据集要比 Cora 和 Citeseer 更加稠密。
- **BlogCatalog.** BlogCatalog[134] 是一个博客用户之间的社交网络。在该数据集中，我们把每个博客用户标注的感兴趣的主题作为其标签。

此外，我们还在一个小型的网络，Zachary's Karate network[135] 上进行了实验，来对我们模型检测出的社区进行可视化。该网络是一个空手道俱乐部成员之间的好友网络，包含 34 个节点和 78 条边。

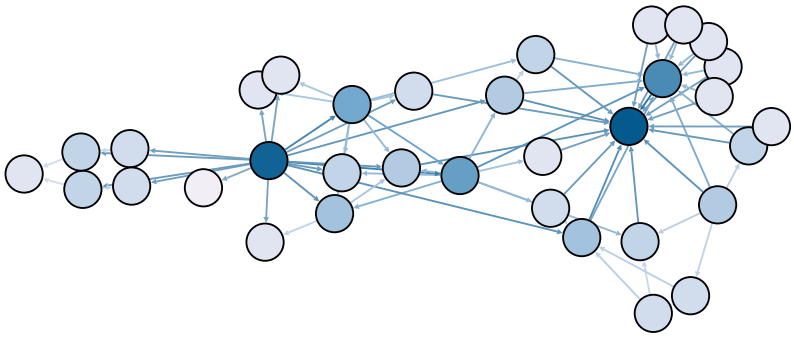


图 7.4　Karate 网络。每个节点颜色的深浅代表了它的度数，颜色越深，表明该节点度数越高。

7.4.2　基准方法

我们采用了 6 个最先进的网络表示学习模型进行对比，包括 **DeepWalk**，**LINE**，**node2vec**，**SDNE**，**MNMF** 以及 **ComE**.。

- **DeepWalk:** DeepWalk[1] 是最典型的网络表示学习方法。它首先在网络上进行随机游走，来得到节点序列。利用这些节点序列，DeepWalk 采用了广泛使用的训练词向量的模型 Skip Gram[5] 来学习网络节点表示。
- **LINE:** LINE[6] 通过对网络节点之间一阶邻近度（1st-order）和二阶邻近度 (2nd-order) 进行建模，来学习大规模网络的节点表示。实验中，我们同时采用了一阶 LINE-1st 和二阶 LINE-2nd 作为基准方法。
- **node2vec:** node2vec[7] 利用 BFS 和 DFS 两种搜索算法，来扩展 DeepWalk 中的随机游走过程，实现了对于网络结构更有效的探索。
- **SDNE:** SDNE[8] 首次采用深层神经网络（自动编码器）来构建网络节点表示。
- **MNMF:** MNMF[104] 通过非负矩阵分解的方式，同时进行非重叠的社区发现和网络表示学习。

- **ComE:** ComE[130] 能够同时学习网络表示和进行重叠社区检测。具体来说，该模型利用一个高维的高斯分布来表示每个社区。

此外，对于链接预测任务，我们还采用了四个典型的基于网络拓扑结构的方法[136]，如下所示：

- **Common Neighbors (CN).** 对于节点 v_i 和 v_j，CN[137] 简单地采用两者共同的邻居来衡量两者相似度。

$$sim(v_i, v_j) = |N_i^+ \cap N_j^+|. \tag{7-13}$$

- **Salton Index.** 对于节点 v_i 和 v_j，Salton index[138] 在 CN 的基础上考虑了两个节点的度数，相似度计算方式如下所示：

$$sim(v_i, v_j) = (|N_i^+ \cap N_j^+|)/(\sqrt{|N_i^+| \times |N_j^+|}). \tag{7-14}$$

- **Jaccard Index.** 对于节点 v_i 和 v_j，Jaccard index 定义如下：

$$sim(v_i, v_j) = (|N_i^+ \cap N_j^+|)/(|N_i^+ \cup N_j^+|). \tag{7-15}$$

- **Resource Allocation Index (RA).** RA index[139] 表示节点 v_j 接收到的资源总量：

$$sim(v_i, v_j) = \sum_{v_k \in N_i^+} \frac{1}{|N_k^+|}. \tag{7-16}$$

对于社区发现任务，我们采用了四个典型的方法作为基准方法：

- **Sequential Clique Percolation (SCP)** [140] 是派系过滤算法 Clique Percolation[124] 的加速版本，通过搜索邻接的派系来检测社区。
- **Link Clustering (LC)**[125] 通过对边进行聚类，来进行社区发现。
- **MDL**[123] 采用最小描述长度原理（minimum description length principle，MDL）来从不同的随机块社区检测模型中进行模型选择。
- **BigCLAM**[127] 是一个典型的基于非负矩阵分解的社区发现方法。将节点的社区分布看作节点表示向量，能够有效地检测出重叠或者嵌套的社区。

7.4.3　评测指标和参数设置

如上文所述，我们在 DeepWalk 和 node2vec 的基础上实现 CNRL 模型。以 DeepWalk 为例，我们将基于统计的分配方法对应的模型记作 S-DW，基于表示的分配方法对应的模型记作 E-DW。同样的，node2vec 对应的 CNRL 模型分别记作 S-n2v 和 E-n2v。

为了进行公平的比较，我们将所有模型的节点表示维度设置为 128。对于 LINE，我们采用原文中的设置，将负例个数设置为 5，学习率设置为 0.025，BlogCatalog 数据集边采样数量为 10^9，其他数据集边采样数量为 10^7。对于随机游走序列，我们设置序列长度为 40，窗口大小为 5，每个节点对应的序列数量为 10。此外，我们采用网格搜索获得 MNMF 模型效果最好的参数设置。

需要注意的是，CNRL 模型中的节点表示包含两部分。因此，为了公平比较，我们将模型中表示向量的初始维度设置为 64，通过拼接可以得到优化后的 128 维的向量表示。此外，平滑因子 α 为 2，β 为 0.5。

对于节点分类任务，Cora、Citeseer 和 Wiki 三个数据集的每个节点只含有一个标签，因此，我们采用 L2 正则逻辑回归分类器 (L2R-LR)[42] 来训练节点分类器。对于 BlogCatalog 中的多标签节点，我们训练 one-vs-rest 逻辑回归分类器，并且采用 *micro-F1* 进行评测。

对于链接预测任务，我们采用标准的评测指标 **AUC**[98]。给定所有节点对之间的相似度，AUC 是指一个随机的未观测到的边的相似度大于随机的不存在的边的相似度的概率。假设我们进行了 n 次独立的比较，AUC 的值为 $(n_1 + 0.5n_2)/n$，其中 n_1 表示为观测到的边有一个更高得分的次数，n_2 表示两者得分相等的次数。

对于社区发现任务，由于这些数据集没有标注真实的社区信息，因此我们采用 **modified modularity**[141] 来衡量检测出的社区的质量。

7.4.4　节点分类

表 7.2、7.3、7.4和 7.5中，我们展示了不同数据集在不同训练比例下的节点分类结果。对于数据集 BlogCataglog，我们采用了更小的训练比例，来加速多标签分类器的训练速度，以及观察 CNRL 模型在稀疏训练数据情形下的效果。从这些表格中，我们观察发现：

- 我们提出的 CNRL 模型在节点分类任务上取得了显著且一致的效果提升。此外，社区优化的 DeepWalk 模型效果优于原始的 DeepWalk 模型，社区优化的 node2vec 模型效果也显著优于原始的 node2vec 模型。这些结果证明了在网络表示学习中引入社区信息的重要性，也验证了 CNRL 模型对于不同模型

表 7.2　Cora 数据集上节点分类准确率 (%)。

% 训练比例	10%	20%	30%	40%	50%	60%	70%	80%	90%
DeepWalk	70.77	73.35	74.53	74.94	75.62	76.07	76.08	76.33	77.27
LINE	70.61	74.79	76.93	77.99	78.66	79.22	79.53	79.35	79.67
node2vec	73.29	76.03	77.52	78.08	78.40	78.51	78.72	79.06	79.15
SDNE	70.97	75.08	76.90	77.82	78.26	79.11	79.37	79.46	79.37
MNMF	75.08	77.85	79.05	79.53	79.82	80.21	79.98	80.11	79.41
ComE	**76.72**	79.25	80.73	80.97	81.53	82.10	82.19	82.42	82.65
S-DW	72.78	75.93	77.47	77.98	78.69	79.14	79.15	78.99	78.23
E-DW	73.35	76.56	77.11	78.63	79.18	79.86	79.96	79.94	80.26
S-n2v	75.86	**79.92**	**81.21**	**82.13**	**82.81**	**83.06**	**82.95**	**83.78**	**83.65**
E-n2v	76.30	79.40	80.62	81.19	81.46	81.82	81.67	82.16	82.80

表 7.3　Citeseer 数据集上节点分类准确率 (%)。

% 训练比例	10%	20%	30%	40%	50%	60%	70%	80%	90%
DeepWalk	47.92	51.54	52.92	54.14	54.21	54.58	55.07	56.09	55.33
LINE	44.27	47.57	50.10	51.15	51.93	52.74	53.46	53.98	54.01
node2vec	49.47	53.27	54.22	55.51	55.87	56.34	56.95	57.61	57.56
SDNE	47.35	51.10	52.45	53.20	53.70	54.20	54.79	55.26	54.46
MNMF	51.62	53.80	55.47	56.94	56.81	57.04	57.05	57.00	57.22
ComE	**54.71**	**57.70**	**58.84**	**59.67**	59.93	60.30	61.12	61.62	61.11
S-DW	49.40	52.58	54.83	55.92	56.63	56.99	57.46	58.48	58.14
E-DW	49.48	52.52	54.41	55.29	56.25	56.21	57.14	57.53	57.41
S-n2v	53.12	56.68	58.20	59.48	**60.31**	**60.76**	**61.21**	**61.90**	**62.63**
E-n2v	51.84	54.33	55.85	56.47	57.19	57.11	57.85	58.67	58.51

表 7.4　Wiki 数据集上节点分类准确率 (%)。

% 训练比例	10%	20%	30%	40%	50%	60%	70%	80%	90%
DeepWalk	58.54	62.12	63.56	65.22	65.90	66.53	67.22	67.50	67.56
LINE	57.53	61.47	63.45	65.14	66.55	67.66	68.35	68.21	68.31
node2vec	58.93	62.60	64.11	65.36	66.03	67.38	67.93	68.26	68.99
SDNE	52.42	57.34	60.15	62.35	63.18	64.21	64.71	65.63	65.60
MNMF	54.76	58.82	60.43	61.66	62.74	63.23	63.46	63.45	64.77
ComE	59.11	62.46	64.38	65.45	65.98	67.38	67.49	67.92	67.89
S-DW	59.97	63.41	65.48	67.03	67.95	69.15	69.45	70.03	70.63
E-DW	58.69	62.37	64.01	65.46	66.16	66.85	66.79	67.05	67.05
S-n2v	**60.66**	**64.43**	**66.63**	**68.23**	**68.92**	**70.52**	**70.41**	**70.62**	**71.60**
E-n2v	60.07	63.81	65.52	66.69	67.64	69.21	69.62	69.40	70.71

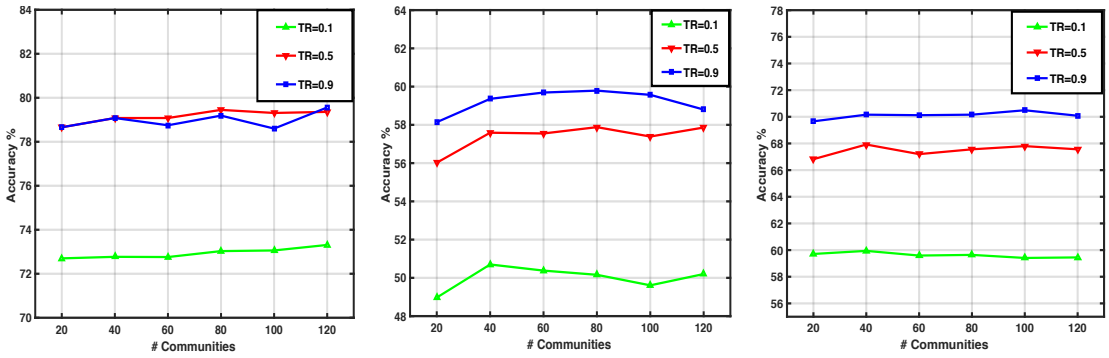
表 7.5　BlogCatalog 数据集上节点分类微平均精确度 (%)。

% 训练比例	1%	2%	3%	4%	5%	6%	7%	8%	9%
DeepWalk	23.66	27.12	28.28	30.02	30.58	31.37	31.57	31.71	32.31
LINE	19.31	23.21	22.88	24.82	25.89	27.00	27.75	28.70	30.04
node2vec	24.47	27.83	29.11	30.61	30.87	31.05	31.50	31.44	31.96
SDNE	17.73	22.38	23.92	25.06	25.65	27.05	27.44	27.72	27.97
MNMF	19.26	21.08	22.29	23.99	25.24	25.97	26.31	26.58	27.16
ComE	22.67	27.43	28.49	29.79	30.34	31.04	31.32	31.58	32.15
S-DW	23.80	27.02	28.63	30.14	30.25	30.96	31.16	31.46	32.45
E-DW	24.93	**28.36**	29.28	**30.80**	**31.19**	31.65	31.72	**32.22**	**32.76**
S-n2v	24.95	27.88	29.17	30.24	30.95	**31.77**	**31.85**	32.12	32.34
E-n2v	**25.75**	28.29	**29.36**	30.54	31.13	31.36	31.67	31.99	32.60

的可扩展性。

- MNMF 模型在 Wiki 和 BlogCatalog 数据集上表现较差，而 CNRL 模型在不同数据集和训练比例下效果稳定。此外，尽管 CNRL 和 MNMF 都结合了社区信息，CNRL 的效果仍然获得了 4% 的提升。这也表明 CNRL 利用网络中社区与文本中主题的类比关系来考虑社区信息方式的合理性。
- CNRL 仅仅只用一半的训练数据，效果就优于几乎所有的基准方法。这表明 CNRL 能够有效的处理标注数据稀疏的问题。

总结来说，我们提出的 CNRL 模型能够有效地将全局的社区信息编码到节点表示中，并且在传统的节点分类任务上获得了显著的提升。此外，CNRL 能够适用于不同类型的网络，无论是稀疏的还是稠密的。最后，与传统的网络表示学习模型相比，CNRL 能够利用更少的标注数据获得更好的效果。



(a) Cora 上分类准确率　(b) Citeseer 上分类准确率　(c) Wiki 上分类准确率

图 7.5　节点分类任务参数敏感性分析 (TR 表示训练比例)。

参数敏感性分析： 为了验证 CNRL 中的社区数目 K 对于节点分类任务的影响，我们在三个数据集上进行参数敏感性实验。这里，我们采用效果最好的 S-n2v

模型，并且设置训练比例（TR）分别为 0.1, 0.5 和 0.9。如图7.5所示，当社区数目发生变化时，CNRL 有着稳定的表现。这表明我们的模型 CNRL 能够适用于检测不同维度的社区，也验证了该模型的鲁棒性。

7.4.5　链接预测

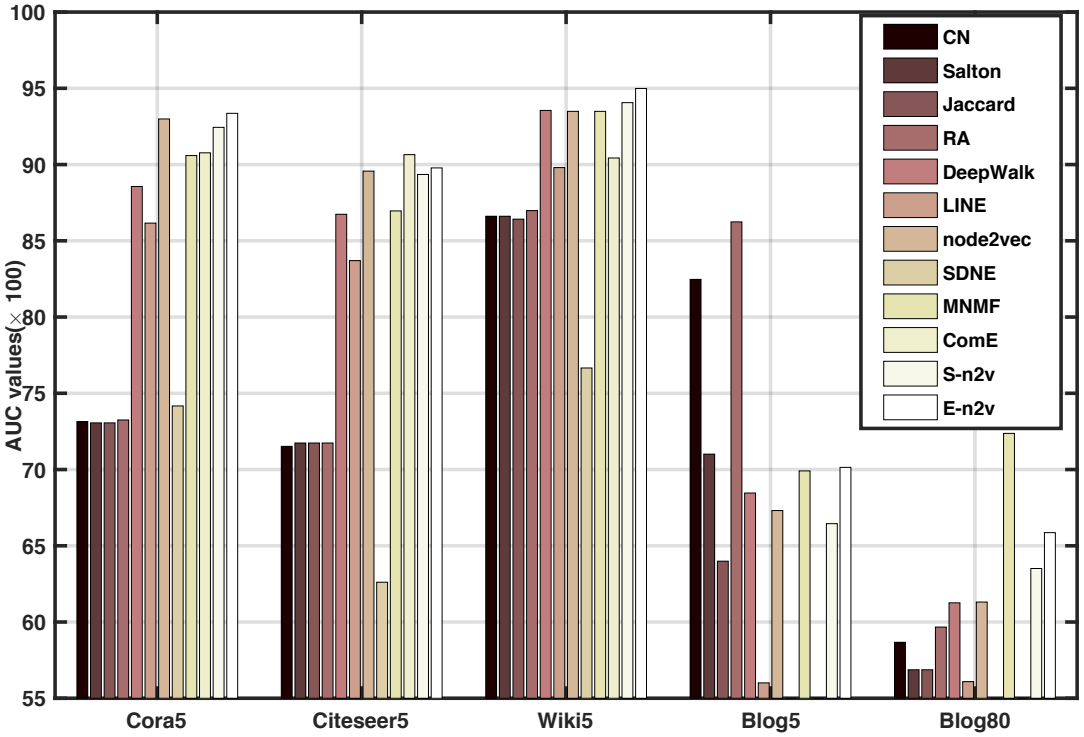
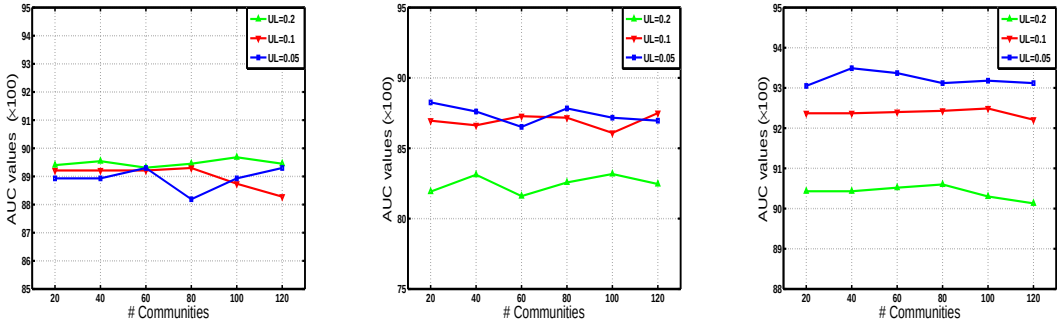


图 7.6　移除不同比例边后的链接预测结果。



(a) Cora 上的 AUC 结果　(b) Citeseer 的 AUC 结果　(c) Wiki 的 AUC 结果

图 7.7　链接预测任务参数敏感性分析 (UL 表示未观测边的比例)。

对于社区进行建模的模型对于边应该有正确预测能力。因此，我们采用了链接预测任务来评测 CNRL 模型效果。

给定一个网络，我们随机去掉一定比例的边作为测试集，剩下的作为训练集。我们用训练集中的边训练节点的网络表示，用这些网络表示来计算节点之间的相似度，进而用来预测节点之间潜在的链接。

为了进行比较，我们采用了两种类型的基准方法，包括基于拓扑结构的链接预测方法和网络表示学习方法。这些方法都需要对节点之间的相似度进行计算。

在图 7.6中，我们展示了移除 5% 的边后，不同模型的 AUC 结果。需要注意的是，在 BlogCatalog 数据集上，由于 LINE-1st 的结果优于 LINE 的结果，我们最终展示 LINE-1st 的结果。从该图中，我们发现：

- 在大多数情形下，网络表示学习的方法效果要优于传统的基于人工定义特征的链接预测方法。这表明网络表示学习方法能够更有效地考虑节点的结构信息并将其用一个低维实值向量表示。在这种情况下，CNRL 的效果要优于大部分的网络表示学习模型。这也验证了 CNRL 考虑社区信息的有效性。
- 对于 BlogCatalog 数据集，节点的平均度数为 32.39，远高于其它数据集节点的平均度数，因此，这种网络结构有利于简单的基于统计的方法，例如 CN 和 RA。然而，如图 7.6所示，当我们移除 80% 的边时，这些简单的基于统计的方法下降明显，接近 25%，然而，CNRL 仅仅下降 5%。这也验证了网络表示学习模型能够处理数据稀疏的情形。

参数敏感性分析： 在链接预测实验上，我们也针对社区数量 K 对于模型效果的影响进行了探究。和节点分类的参数敏感性分析一致，我们同样采用 S-n2v 模型，将社区数量 K 从 20 增加到 120。如图 7.7所示，随着社区数量的变化，CNRL 的效果依然非常稳定，AUC 结果在所有数据集上变化误差不超过 2%。敏感性分析结果表明，CNRL 对于超参数 K 不敏感，能够方便地适用于不同场景。

7.4.6　社区检测

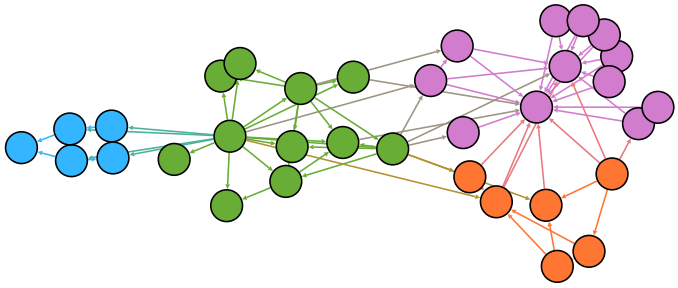
表 7.6　社区检测结果。

数据集	SCP	LC	MDL	BigCLAM	S-DW	E-DW	S-n2v	E-n2v
Cora	0.076	0.334	0.427	0.464	0.464	**1.440**	0.447	1.108
Citeseer	0.055	0.315	0.439	0.403	0.486	**1.861**	0.485	1.515
Wiki	0.063	0.322	0.300	0.286	0.291	**0.564**	0.260	0.564

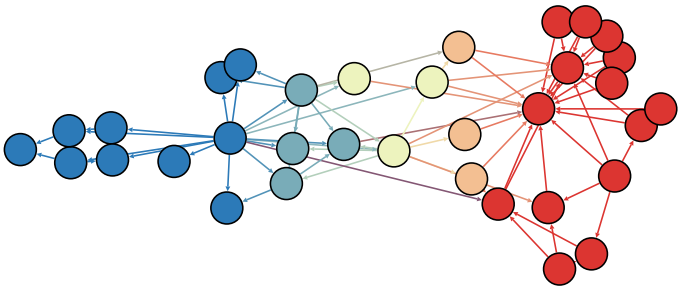
由于 BigCLAM 和 CNRL 得到的是每个节点在不同社区上的概率分布，而不是每个节点属于哪些社区，我们简单的设置一个阈值 $\tau = 0.1$ 来选择每个节点所属的社区。从表 7.6中，我们发现，S-CNRL 方法 (S-DW or S-n2v) 和其它最先进的社区发现算法效果可比，而 E-CNRL 方法 (E-DW or E-n2v) 的效果显著地优于其它所

有方法。这表明 CNRL 能够进行高质量的社区检测，也验证了我们将网络中的社区类比成文本中的主题的合理性。

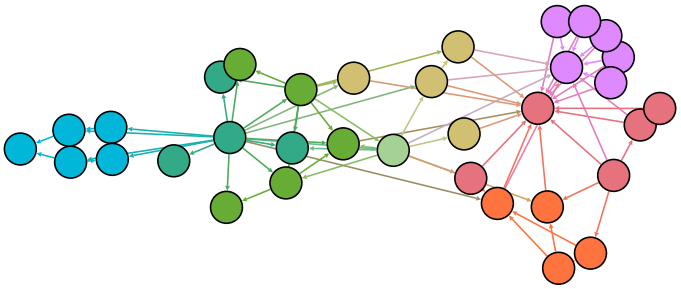
7.4.7　社区检测



(a) 快速折叠算法（Fast Unfolding）。



(b) CNRL 的 2 个社区。



(c) CNRL 的 4 个社区。

图 7.8　社区检测结果可视化。

为了更直观地展示检测出来的社区，我们在一个小型的空手道网络上展示 CNRL 进行重叠社区检测的结果。为了进行对比，我们同样展示一个经典的非重叠社区检测方法 Fast Unfolding[142] 的检测结果。在图 7.8中，我们用不同的颜色标注不同的社区，用渐变色来表示属于多个社区的节点。从该图中，我们发现：

- CNRL 能够有效地检测出不同规模的社区结构，而不是简单地将所有节点进行社区分割。2 个社区的检测结果和 4 个社区的检测结果都能够很好符合该网络的结构。

- CNRL 能够很好地处理社区检测中的重叠问题。它能够准确地检测出社区边界上的节点，并平衡该点属于不同社区的权重。

此外，在 CNRL 中，我们假设节点序列包含了全局的社区特征，网络的社区可以通过这些节点序列检测出来。图 7.8中的结果符合我们的直观认知，也验证了我们的猜想。

7.4.8　示例

为了更直观地展示 CNRL 的有效性和解释 CNRL 检测出的社区含义，我们选取了三个有代表性的示例。

7.4.8.1　最近邻

表 7.7　论文 “Protein Secondary Structure Modelling with Probabilistic Networks’’ 的最近邻。

CNRL	
Using Dirichlet Mixture Priors to Derive Hidden Markov Models for Protein Families	Neural Networks
Optimal Alignments in Linear Space using Automaton-derived Cost Functions	Neural Networks
Dirichlet Mixtures: A Method for Improving Detection of Weak but Significant Protein Sequence Homology	Neural Networks
Family-based Homology Detection via Pairwise Sequence Comparison	Neural Networks
The megaprior heuristic for discovering protein sequence patterns	Neural Networks
DeepWalk	
An Optimal Weighting Criterion of Case Indexing for Both Numeric and Symbolic Attributes	Case Based
Using Dirichlet Mixture Priors to Derive Hidden Markov Models for Protein Families	Neural Networks
On The State of Evolutionary Computation	Genetic Algorithms
Optimal Alignments in Linear Space using Automaton-derived Cost Functions	Neural Networks
On Biases in Estimating Multi-Valued Attributes	Rule Learning

我们从 Cora 数据集中选取一个例子，来说明在网络表示学习中考虑社区信息的重要性。该论文节点的题目为 “Protein Secondary Structure Modelling with Probabilistic Networks’’，属于 “Neural Networks’’ 领域。我们分别利用 CNRL 和 DeepWalk 学习到的节点表示，求出该论文的 top-5 最近邻节点，如图 7.7所示。

从该表中，最直观的观察结果是，CNRL 推荐的 5 个邻居与当前节点拥有相同的标签，而 DeepWalk 仅有 2 个邻居拥有相同的标签。这个结果是合理的，因为同一个社区中的节点倾向于共享同样的属性，这些共享的属性通常隐藏在社区中，从而被 CNRL 所利用。

通过观察每个节点的题目，我们发现 DeepWalk 找到的大部分邻居节点与当前节点在主题上没有关系，而 CNRL 找到的邻居节点与当前节点在主题上更相关，这也验证了同一个网络社区中的节点在属性或主题上存在联系。

7.4.8.2　社区分配

表 7.8　社区分配结果。

代表节点	类别
Community 1 (weight = 0.56)	
Learning to Act using Real-Time Dynamic Programming	Reinforcement Learning
Generalized Markov Decision Processes: Dynamic-programming and Reinforcement-learning Algorithms	Reinforcement Learning
On the Convergence of Stochastic Iterative Dynamic Programming Algorithms	Reinforcement Learning
Community 2 (weight = 0.20)	
The Structure-Mapping Engine: Algorithm and Examples	Case Based
Case-based reasoning: Foundational issues, methodological variations, and system approaches	Case Based
Concept Learning and Heuristic Classification in Weak-Theory Domains	Case Based
Community 3 (weight = 0.12)	
Learning to Predict by the Methods of Temporal Differences	Reinforcement Learning
Generalization in Reinforcement Learning: Safely Approximating the Value Function	Reinforcement Learning
Exploration and Model Building in Mobile Robot Domains	Reinforcement Learning

与 DeepWalk、node2vec 等模型相比，CNRL 不仅能学习到社区优化的节点表示，也能够得到每个节点的社区分布。为了更直观的说明节点的社区分布，我们从 Cora 中选取一个示例节点，并在表 7.8中展示它的社区分布结果。被选取的论文节点题目为 “Using a Case Base of Surfaces to Speed-Up Reinforcement Learning’’，该节点属于 “Reinforcement Learning’’ 类别。对于每个社区，我们通过公式 (7-6) 来选取最有代表性的节点。

从该表中，我们发现，CNRL 检测出的每个社区都有其各自的特点。例如，社

区 1 与 “Dynamic Programming’’ 相关，这是 “Reinforcement Learning’’ 的一个子领域；而社区 2 与 “Cased Based’’ 研究相关；社区 3 更关注 “Reinforcement Learning’’ 中的学习和建模方法。

根据这些选取的节点的题目，我们发现当前节点确实与这些所有的社区都相关，而每个社区的权重也反映了该节点与它们的相关程度。

7.4.8.3　全局社区特征

表 7.9　Cora 中社区的代表性词语。

社区编号	典型词和对应词频
0	Models:13 Hidden:11 Markov:10
6	Reasoning:13 Case-Based:13 Knowledge:7
8	Genetic:23 Programming:16 Evolution:9
13	Boosting:6 Bayesian:6 Classifiers:6
15	Neural:14 Networks:11 Constructive:7
19	Reinforcement:21 Markov:8 Decision:8

在这一小节中，我们尝试从全局的角度来解释 CNRL 发现的全局的社区特征。我们设置社区数量 $K = 20$，在 Cora 上训练 S-DW 模型，然后根据公式 (7-6) 选取每个社区最有代表性的节点。我们保留 $\Pr(v|c) > 0.005$ 的节点。随后，我们统计出这些节点对应的题目中最高频的词语，作为该社区的代表性词语。由于 Cora 是一个机器学习论文数据集，大多数论文节点的题目中包含 “Machine’’ 和 “Learning’’ 两词，因此我们去掉了这两个词。由于空间限制，我们只在表 7.9中列举出最有代表性的 6 个社区。

从该表中，我们发现，我们能够根据这些社区信息，对机器学习领域有一个初步的认知。这些社区具有很好的区分性，分别对应着 ‘Hidden Markov Models”, “Case-based Reasoning”, “Genetic Programming”, “Neural Networks’’ 和 “Reinforcement Learning’’ 等。这表明，CNRL 能够有效地检测出高质量的社区。

7.5　本章小结

在本章工作中，我们提出一个新颖的社区优化的网络表示学习框架，来利用社区这个网络中关键的全局特征，提高网络节点表示质量。在该框架中，我们能够同时学习网络节点表示和进行社区检测，检测出来的社区结果也会影响节点表示的学习。此外，CNRL 能够适用于不同的基于随机游走的网络表示学习模型，例

如 DeepWalk 和 node2vec。在多个网络分析任务上，CNRL 都比已有的网络表示学习模型取得了显著的效果提升。

第 8 章　总结与展望

网络节点表示是网络分析任务的基础，在数据挖掘和社交网络分析等领域有着重要的作用。传统的网络表示学习方法分为基于符号的显式表示和基于表示学习的隐式表示，这些已有的方法一般仅仅根据网络节点的结构信息来得到节点的表示向量，而忽略实际的社交网络场景中丰富的异构信息。此外，这些方法也面临着可解释性问题和计算效率问题。本文分别从网络节点的显式、隐式表示两个角度出发，探究融合异构信息的网络表示方法，来提高一系列网络分析任务的效果，并改进它们的可解释性和计算效率。

8.1　论文的主要贡献

总结来说，本文主要包括以下几点主要贡献.

首先，针对网络节点的显式表示，我们充分考虑实际的社交网络场景，提出了基于词项的显式表示以及基于标签主题的显式表示两种方案。这些方法能够有效地融合真实场景中的多源异构信息，探究这些多源异构信息与用户属性、标签之间的关系。在用户属性预测以及社会标签推荐任务上，我们验证了这两种表示方案的有效性。

其次，针对网络节点的隐式表示，我们充分利用已有的表示学习技术，提出了一系列融合异构信息的网络表示学习方案。具体来说，我们针对节点的类别标签信息，利用最大间隔理论，提出 MMDW 模型，将这些标签信息融入到网络表示学习过程中，来得到更有区分性的网络节点表示，并在节点分类任务上验证了这种解决方案的效果。此外，我们针对节点附加的文本信息，对节点之间的动态关系进行建模，提出上下文相关的网络表示学习方案 CANE，并在链接预测任务上验证了上下文相关的网络表示对于节点之间的链接关系进行建模的能力。针对网络节点之间边上丰富的语义信息，我们提出基于平移思想的网络表示学习模型 TransNet，来对节点表示及边的表示之间的关系进行建模。对于未知关系的节点对，该模型能够预测它们之间边上的标签信息，也就是进行社会关系抽取。通过社会关系抽取任务，验证了 TransNet 模型对于节点之间显式的关系的建模能力。最后，针对网络结构中全局的社区特征，我们提出了社区优化的网络表示学习模型 CNRL，利用网络中的社区与文本中的主题之间的类比关系，来同时进行社区检测以及节点表示学习，该模型的效果在一系列网络分析任务中得到了验证。

上述提出的方法有针对性地对不同的异构信息进行了融合，并有效解决了相

应的社交网络分析任务。

8.2　工作展望

最后，网络表示学习领域仍存在一些重要问题亟待探索和解决，包括：

- **大规模网络表示学习**：目前已有的网络表示方法一般仅能够对有限规模的网络或者社会网络中的子网络进行节点表示的学习，而不能够处理实际场景中的上亿个节点社交网络。如何有效地对这种大规模的网络进行训练，是网络表示学习在实际场景中有效应用的基础。
- **网络表示模型压缩**：针对大规模社交网络的训练，除了训练时间上的开销，模型占用的空间上的开销也不可忽视。举例来说，仅仅考虑网络表示学习模型中的网络节点表示的参数，1 亿个节点 200 维表示向量占用内存就接近 200G。如果再考虑复杂的异构信息，或者使用更复杂的基于神经网络的模型，模型的参数规模会进一步增加。如何有效地利用网络中节点表示之间的关系以及节点的长尾分布等规律，对网络规模进行压缩，也影响着网络表示学习在实际场景中的应用。
- **动态网络表示学习**：真实世界中的网络，其中的网络结构一直在动态变化。在社交网络中，这种动态变化也反映出网络节点的属性、兴趣的变化。而目前已有的模型基本上都是针对固定的网络结构来学习节点表示。针对动态网络，实现在线即时的网络表示学习训练和更新，也是重要的研究问题。
- **引入知识的网络表示学习**：目前基于隐式表示的网络表示学习模型，没有考虑到知识图谱等固定结构的外部知识信息，而这种外部结构化的知识能够为网络表示学习及后续的网络分析应用引入可解释性及推理能力。因此，将外部结构化的知识图谱信息引入社交网络分析以及网络表示学习中，也是具有巨大潜力的研究方向。
- **面向具体应用的网络表示学习**：目前已有的网络表示学习模型的训练目标，一般侧重于对于网络结构的建模和重构能力，而忽略学习到的表示向量在后续应用场景中的效果。因此，如何将网络表示学习模型与具体的应用场景相结合，有针对性的改善节点表示在具体应用中的效果，是网络表示学习在应用方面重要的挑战。

总之，网络表示是进行用户画像、行为分析、异常检测、推荐系统等实际场景应用的基础工具。通过解决上述问题，可以有效解决网络表示在实际场景中的有效利用和效果提升问题。由此可见，针对网络表示的研究，无论从理论研究方面，还是从实际应用方面，都具有极大的探索空间和研究价值。

参考文献

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

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

[3] Graves A, Mohamed A r, Hinton G. Speech recognition with deep recurrent neural networks [C]//Proceedings of ICASSP. 2013: 6645–6649.

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

[5] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[C]//Proceedings of ICIR. 2013.

[6] Tang J, Qu M, Wang M, et al. Line: Large-scale information network embedding[C]// Proceedings of WWW. 2015: 1067–1077.

[7] Grover A, Leskovec J. Node2vec: Scalable feature learning for networks[C]//Proceedings of KDD. 2016.

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

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

[10] Yang C, Liu Z, Zhao D, et al. Network representation learning with rich text information[C]// Proceedings of IJCAI. 2015: 2111–2117.

[11] Qiu J, Dong Y, Ma H, et al. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec[C]//Proceedings of WSDM. 2018: 459–467.

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

[13] Burger J D, Henderson J, Kim G, et al. Discriminating gender on twitter[C]//Proceedings of EMNLP. 2011: 1301–1309.

[14] Schwartz H A, Eichstaedt J C, Kern M L, et al. Personality, gender, and age in the language of social media: The open-vocabulary approach[J]. PloS one, 2013, 8(9): e73791.

[15] Dodds P S, Harris K D, Kloumann I M, et al. Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter[J]. PLoS ONE, 2011, 6(12): e26752.

[16] Rao D, Yarowsky D, Shreevats A, et al. Classifying latent user attributes in twitter[C]// Proceedings of Workshop on SMUC. 2010: 37–44.

[17] Newman M E. Modularity and community structure in networks[J]. PNAS, 2006, 103(23): 8577–8582.

[18] McPherson M, Smith-Lovin L, Cook J M. Birds of a feather: Homophily in social networks[J]. Annual Review of Sociology, 2001, 27: 415–444.

[19] Rothman R A. Working: Sociological perspectives[M]. [S.l.]: Prentice-Hall Englewood Cliffs, NJ, 1987

[20] Volti R. An introduction to the sociology of work and occupations[M]. [S.l.]: Pine Forge Press, 2011

[21] Holland J L. Making vocational choices: A theory of vocational personalities and work environments[M]. [S.l.]: Psychological Assessment Resources, 1997

[22] Lazer D, Pentland A S, Adamic L, et al. Life in the network: The coming age of computational social science[J]. Science, 2009, 323(5915): 721.

[23] Kosinski M, Stillwell D, Graepel T. Private traits and attributes are predictable from digital records of human behavior[J]. PNAS, 2013, 110(15): 5802–5805.

[24] Lewis K, Gonzalez M, Kaufman J. Social selection and peer influence in an online social network[J]. PNAS, 2012, 109(1): 68–72.

[25] Mislove A, Lehmann S, Ahn Y Y, et al. Understanding the demographics of twitter users[C]// Proceedings of ICWSM. 2011.

[26] Goswami S, Sarkar S, Rustagi M. Stylometric analysis of bloggers' age and gender[C]// Proceedings of ICWSM. 2009.

[27] Fink C, Kopecky J, Morawski M. Inferring gender from the content of tweets: A region specific example[C]//Proceedings of ICWSM. 2012.

[28] Li R, Wang S, Deng H, et al. Towards social user profiling: Unified and discriminative influence model for inferring home locations[C]//Proceedings of KDD. 2012: 1023–1031.

[29] Feng W, Wang J. Incorporating heterogeneous information for personalized tag recommendation in social tagging systems[C]//Proceedings of KDD. 2012: 1276–1284.

[30] Liu Z, Tu C, Sun M. Tag dispatch model with social network regularization for microblog user tag suggestion[C]//Proceedings of Coling. 2012: 755.

[31] Tu C, Liu Z, Sun M. Inferring correspondences from multiple sources for microblog user tags [C]//Chinese National Conference on Social Media Processing. [S.l.]: Springer, 2014: 1–12.

[32] Kong X, Cao B, Yu P S. Multi-label classification by mining label and instance correlations from heterogeneous information networks[C]//Proceedings of KDD. 2013: 614–622.

[33] Chaudhari G, Avadhanula V, Sarawagi S. A few good predictions: selective node labeling in a social network[C]//Proceedings of WSDM. 2014: 353–362.

[34] Jacob Y, Denoyer L, Gallinari P. Learning latent representations of nodes for classifying in heterogeneous social networks[C]//Proceedings WSDM. 2014: 373–382.

[35] Mislove A, Viswanath B, Gummadi K P, et al. You are who you know: inferring user profiles in online social networks[C]//Proceedings of WSDM. 2010: 251–260.

[36] Yang S H, Long B, Smola A, et al. Like like alike: Joint friendship and interest propagation in social networks[C]//Proceedings of WWW. 2011: 537–546.

[37] Sachan M, Dubey A, Srivastava S, et al. Spatial compactness meets topical consistency: Jointly modeling links and content for community detection[C]//Proceedings of WSDM. 2014: 503–512.

[38] Golbeck J, Robles C, Turner K. Predicting personality with social media[C]//Proceedings of CHI. 2011: 253–262.

[39] Danescu-Niculescu-Mizil C, Lee L, Pang B, et al. Echoes of power: Language effects and power differences in social interaction[C]//Proceedings of WWW. 2012: 699–708.

[40] Yang Y, Pedersen J O. A comparative study on feature selection in text categorization[C]// Proceedings of ICML: volume 97. 1997: 412–420.

[41] Forman G. An extensive empirical study of feature selection metrics for text classification[J]. JMLR, 2003, 3: 1289–1305.

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

[43] Zhu X, Goldberg A B. Introduction to semi-supervised learning[J]. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2009, 3(1): 1–130.

[44] Chafetz J S. The gender division of labor and the reproduction of female disadvantage toward an integrated theory[J]. Journal of Family Issues, 1988, 9(1): 108–131.

[45] Niederhoffer K G, Pennebaker J W. Linguistic style matching in social interaction[J]. Journal of Language and Social Psychology, 2002, 21(4): 337–360.

[46] Mairesse F, Walker M A, Mehl M R, et al. Using linguistic cues for the automatic recognition of personality in conversation and text[J]. JAIR, 2007, 30: 457–500.

[47] Liang H, Xu Y, Li Y, et al. Connecting users and items with weighted tags for personalized item recommendations[C]//Proceedings of HT. 2010: 51–60.

[48] Peng J, Zeng D, Zhao H, et al. Collaborative filtering in social tagging systems based on joint item-tag recommendations[C]//Proceedings of CIKM. 2010: 809–818.

[49] Zhen Y, Li W, Yeung D. Tagicofi: tag informed collaborative filtering[C]//Proceedings of RecSys. 2009: 69–76.

[50] Symeonidis P, Nanopoulos A, Manolopoulos Y. Tag recommendations based on tensor dimensionality reduction[C]//Proceedings of RecSys. 2008: 43–50.

[51] Rendle S, Balby Marinho L, Nanopoulos A, et al. Learning optimal ranking with tensor factorization for tag recommendation[C]//Proceedings of KDD. 2009: 727–736.

[52] Rendle S, Schmidt-Thieme L. Pairwise interaction tensor factorization for personalized tag recommendation[C]//Proceedings of WSDM. 2010: 81–90.

[53] Jaschke R, Marinho L, Hotho A, et al. Tag recommendations in social bookmarking systems[J]. AI Communications, 2008, 21(4): 231–247.

[54] Ohkura T, Kiyota Y, Nakagawa H. Browsing system for weblog articles based on automated folksonomy[C]//Proceedings of WWW. 2006.

[55] Mishne G. Autotag: a collaborative approach to automated tag assignment for weblog posts [C]//Proceedings of WWW. 2006: 953–954.

[56] Lee S, Chun A. Automatic tag recommendation for the web 2.0 blogosphere using collaborative tagging and hybrid ann semantic structures[C]//Proceedings of ACS. 2007: 88–93.

[57] Katakis I, Tsoumakas G, Vlahavas I. Multilabel text classification for automated tag suggestion [C]//Proceedings of ECML-PKDD: volume 18. 2008.

[58] Fujimura S, Fujimura K, Okuda H. Blogosonomy: Autotagging any text using bloggers' knowledge[C]//Proceedings of WI. 2007: 205–212.

[59] Heymann P, Ramage D, Garcia-Molina H. Social tag prediction[C]//Proceedings of SIGIR. 2008: 531–538.

[60] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. JMLR, 2003, 3: 993–1022.

[61] Krestel R, Fankhauser P, Nejdl W. Latent dirichlet allocation for tag recommendation[C]// Proceedings of RecSys. 2009: 61–68.

[62] Si X, Sun M. Tag-LDA for scalable real-time tag recommendation[J]. JCIS, 2009, 6(1): 23–31.

[63] Bundschus M, Yu S, Tresp V, et al. Hierarchical bayesian models for collaborative tagging systems[C]//Proceedings of ICDM. 2009: 728–733.

[64] Iwata T, Yamada T, Ueda N. Modeling social annotation data with content relevance using a topic model[C]//Proceedings of NIPS. 2009: 835–843.

[65] Blei D, Jordan M. Modeling annotated data[C]//Proceedings of SIGIR. 2003: 127–134.

[66] Griffiths T L, Steyvers M. Finding scientific topics[J]. PNAS, 2004.

[67] Gregor H. Parameter estimation for text analysis[J]. Technical report, 2005.

[68] Andrieu C, De Freitas N, Doucet A, et al. An introduction to mcmc for machine learning[J]. Machine learning, 2003, 50(1): 5–43.

[69] Manning C D, Raghavan P, Schütze H. Introduction to information retrieval: volume 1[M]. [S.l.]: Cambridge University Press, 2008

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

[71] Chang J, Blei D M. Relational topic models for document networks[C]//Proceedings of AIStats. 2009: 81–88.

[72] Cohn D, Chang H. Learning to probabilistically identify authoritative documents[C]// Proceedings of ICML. 2000: 167–174.

[73] Hearst M A, Dumais S T, Osman E, et al. Support vector machines[J]. IEEE Intelligent Systems and their Applications, 1998, 13(4): 18–28.

[74] Roller B T C G D. Max-margin markov networks[C]//Proceedings of NIPS. 2004.

[75] Zhu J, Ahmed A, Xing E P. Medlda: maximum margin supervised topic models[J]. JMLR, 2012, 13(1): 2237–2278.

[76] Pei W, Ge T, Chang B. Max-margin tensor neural network for chinese word segmentation.[C]// Proceedings of ACL. 2014: 293–303.

[77] Taskar B, Klein D, Collins M, et al. Max-margin parsing.[C]//Proceedings of EMNLP: volume 1. 2004: 3.

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

[79] Yu H F, Jain P, Kar P, et al. Large-scale multi-label learning with missing labels[C]//Proceedings of ICML. 2014: 593–601.

[80] Crammer K, Singer Y. On the learnability and design of output codes for multiclass problems [J]. Machine learning, 2002, 47(2-3): 201–233.

[81] Keerthi S S, Sundararajan S, Chang K W, et al. A sequential dual method for large scale multi-class linear svms[C]//Proceedings of KDD. 2008: 408–416.

[82] McCallum A, Nigam K, Rennie J, et al. Automating the construction of internet portals with machine learning[J]. Information Retrieval Journal, 2000, 3: 127–163.

[83] Sen P, Namata G M, Bilgic M, et al. Collective classification in network data[J]. AI Magazine, 2008, 29(3): 93–106.

[84] Blunsom P, Grefenstette E, Kalchbrenner N. A convolutional neural network for modelling sentences[C]//Proceedings of ACL. 2014.

[85] Johnson R, Zhang T. Effective use of word order for text categorization with convolutional neural networks[C]//Proceedings of NAACL. 2015.

[86] Kim Y. Convolutional neural networks for sentence classification[C]//Proceedings of EMNLP. 2014.

[87] Kiros R, Zhu Y, Salakhutdinov R R, et al. Skip-thought vectors[C]//Proceedings of NIPS. 2015: 3294–3302.

[88] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks[C]//Proceedings of ACL. 2015.

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

[90] Chen J, Zhang Q, Huang X. Incorporate group information to enhance network embedding[C]// Proceedings of CIKM. 2016.

[91] Sun X, Guo J, Ding X, et al. A general framework for content-enhanced network representation learning[J]. arXiv preprint arXiv:1610.02906, 2016.

[92] Schuster M, Paliwal K K. Bidirectional recurrent neural networks[J]. IEEE TSP, 1997, 45(11): 2673–2681.

[93] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation[C]//Proceedings of EMNLP. 2014.

[94] Hu B, Lu Z, Li H, et al. Convolutional neural network architectures for matching natural language sentences[C]//Proceedings of NIPS. 2014: 2042–2050.

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

[96] Leskovec J, Kleinberg J, Faloutsos C. Graphs over time: densification laws, shrinking diameters and possible explanations[C]//Proceedings of KDD. 2005: 177–187.

[97] Airoldi E M, Blei D M, Fienberg S E, et al. Mixed membership stochastic blockmodels[J]. JMLR, 2008, 9: 1981–2014.

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

[99] Lindamood J, Heatherly R, Kantarcioglu M, et al. Inferring private information using social network data[C]//Proceedings of WWW. 2009: 1145–1146.

[100] Shepitsen A, Gemmell J, Mobasher B, et al. Personalized recommendation in social tagging systems using hierarchical clustering[C]//Proceedings of RecSys. 2008.

[101] Liben-Nowell D, Kleinberg J. The link-prediction problem for social networks[J]. JASIST, 2007.

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

[103] Bordes A, Usunier N, Garcia-Duran A, et al. Translating embeddings for modeling multi-relational data[C]//Proceedings of NIPS. 2013: 2787–2795.

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

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

[106] Riedel S, Yao L, McCallum A. Modeling relations and their mentions without labeled text[C]// Proceedings of ECML-PKDD. 2010: 148–163.

[107] Hoffmann R, Zhang C, Ling X, et al. Knowledge-based weak supervision for information extraction of overlapping relations[C]//Proceedings of ACL. 2011: 541–550.

[108] Surdeanu M, Tibshirani J, Nallapati R, et al. Multi-instance multi-label learning for relation extraction[C]//Proceedings of EMNLP. 2012: 455–465.

[109] Lin Y, Shen S, Liu Z, et al. Neural relation extraction with selective attention over instances [C]//Proceedings of ACL: volume 1. 2016: 2124–2133.

[110] Bollacker K, Evans C, Paritosh P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]//Proceedings of SIGMOD. 2008: 1247–1250.

[111] Auer S, Bizer C, Kobilarov G, et al. Dbpedia: A nucleus for a web of open data[M]//The semantic web. 2007: 722–735

[112] Suchanek F M, Kasneci G, Weikum G. Yago: a core of semantic knowledge[C]//Proceedings of WWW. 2007: 697–706.

[113] Srivastava N, Hinton G E, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting.[J]. JMLR, 2014, 15(1): 1929–1958.

[114] Tang J, Zhang J, Yao L, et al. Arnetminer: Extraction and mining of academic social networks [C]//Proceedings of KDD. 2008: 990–998.

[115] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in python[J]. JMLR, 2011, 12: 2825–2830.

[116] Yang J, McAuley J, Leskovec J. Community detection in networks with node attributes[C]// Proceedings of ICDM. 2013.

[117] Yang B, Liu D, Liu J. Discovering communities from social networks: Methodologies and applications[M]//Handbook of social network technologies and applications. 2010: 331–346

[118] Fortunato S. Community detection in graphs[J]. Physics reports, 2010, 486(3): 75–174.

[119] Kernighan B W, Lin S. An efficient heuristic procedure for partitioning graphs[J]. The Bell system technical journal, 1970, 49(2): 291–307.

[120] Newman M E, Girvan M. Finding and evaluating community structure in networks[J]. Physical review E, 2004, 69(2): 026113.

[121] Pothen A, Simon H D, Liou K P. Partitioning sparse matrices with eigenvectors of graphs[J]. SIAM journal on matrix analysis and applications, 1990, 11(3): 430–452.

[122] Nowicki K, Snijders T A B. Estimation and prediction for stochastic blockstructures[J]. JASA, 2001, 96(455): 1077–1087.

[123] Peixoto T P. Model selection and hypothesis testing for large-scale network models with overlapping groups[J]. Physical Review X, 2015, 5(1): 011033.

[124] Palla G, Derényi I, Farkas I, et al. Uncovering the overlapping community structure of complex networks in nature and society[J]. Nature, 2005, 435(7043): 814–818.

[125] Ahn Y Y, Bagrow J P, Lehmann S. Link communities reveal multiscale complexity in networks [J]. Nature, 2010.

[126] Wang F, Li T, Wang X, et al. Community discovery using nonnegative matrix factorization[J]. Data Mining and Knowledge Discovery, 2011.

[127] Yang J, Leskovec J. Community-affiliation graph model for overlapping network community detection[C]//Proceedings of ICDM. 2012.

[128] Xie J, Kelley S, Szymanski B K. Overlapping community detection in networks: The state-of-the-art and comparative study[J]. CSUR, 2013.

[129] Yang J, Leskovec J. Overlapping community detection at scale: a nonnegative matrix factorization approach[C]//Proceedings of WSDM. 2013.

[130] Cavallari S, Zheng V W, Cai H, et al. Learning community embedding with community detection and node embedding on graphs[C]//Proceedings of CIKM. 2017.

[131] Wang Y, Bai H, Stanton M, et al. Plda: Parallel latent dirichlet allocation for large-scale applications[C]//Proceedings of AAIM. 2009: 301–314.

[132] Liu Z, Zhang Y, Chang E Y, et al. Plda+: Parallel latent dirichlet allocation with data placement and pipeline processing[J]. ACM TIST, 2011, 2(3): 26.

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

[134] Tang L, Liu H. Relational learning via latent social dimensions[C]//Proceedings of KDD. 2009.

[135] Zachary W W. An information flow model for conflict and fission in small groups[J]. JAR, 1977.

[136] Lü L, Zhou T. Link prediction in complex networks: A survey[J]. Physica A, 2011.

[137] Newman M E. Clustering and preferential attachment in growing networks[J]. Physical Review E, 2001.

[138] Salton G, McGill M J. Introduction to modern information retrieval[M]. 1986.

[139] Zhou T, Lü L, Zhang Y C. Predicting missing links via local information[J]. EPJ B, 2009.

[140] Kumpula J M, Kivelä M, Kaski K, et al. Sequential algorithm for fast clique percolation[J]. Physical Review E, 2008.

[141] Zhang H, King I, Lyu M R. Incorporating implicit link preference into overlapping community detection[C]//Proceedings of AAAI. 2015.

[142] Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks [J]. JSTAT, 2008.

致　谢

衷心感谢我的导师孙茂松教授对我的悉心指导，您的严谨治学的态度一直是我的学习榜样。也衷心感谢我的副指导老师刘知远副教授，在过去 7 年的本科和博士生涯中，您一直是我的良师益友，在我的生活和科研的低谷时给予了我莫大的帮助，也对我今后的职业规划提供了巨大的支持，在您的指导和帮助下我才能一直奋斗到今天。感谢我的父亲和母亲，你们对我一如既往的信任、宽容和鼓励是我永远最踏实的后盾和前进的动力。感谢刘洋老师、栾焕博老师和其他曾经给予我帮助的各位老师。感谢我的各位合作者，张惟诚、刘晗、张正彦、王豪、曾祥楷、龙上邦、宋长河、郭志芃、钟皓曦、李想、胡紫昆等同学，和你们宝贵的合作经历让我能够在科研的探索道路上不断前行。感谢以前和现在研究小组的各位本科生同学，你们的共同努力才能让我们研究小组一直欣欣向荣。此外，我要特别感谢实验室所有伙伴们，柳春洋、沈世奇、阿雅娜、张檬、林衍凯、杨成、谢若冰、陈翱、陈慧敏、张菡、张嘉成、刘正皓、丁延卓、梁健楠、矣晓沅、王宇星、尹向荣等，谢谢你们对我的帮助、鼓励和陪伴，与你们朝夕相伴的欢乐时光是我的博士生涯最珍贵的回忆。感谢曾经的和现在的所有帮助、指导过我的同学和朋友。

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

1990 年 2 月 20 日出生于山东省滕州市。

2009 年 9 月考入清华大学计算机科学与技术系计算机科学与技术专业，2013 年 7 月本科毕业并获得工学学士学位。

2013 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] **Cunchao Tu**, Han Liu, Zhiyuan Liu, Maosong Sun. CANE: Context-Aware Network Embedding for Relation Modeling. Annual Meeting of the Association for Computational Linguistics (ACL'17), 2017. **(CCF A)**

[2] **Cunchao Tu**, Zhengyan Zhang, Zhiyuan Liu, Maosong Sun. TransNet: Translation-Based Network Representation Learning for Social Relation Extraction. International Joint Conference on Artificial Intelligence (IJCAI'17), 2017. **(CCF A)**

[3] **Cunchao Tu**, Weicheng Zhang, Zhiyuan Liu, Maosong Sun. Max-Margin DeepWalk: Discriminative Learning of Network Representation. International Joint Conference on Artificial Intelligence (IJCAI'16), 2016. **(CCF A)**

[4] **Cunchao Tu**, Xiangkai Zeng, Hao Wang, Zhiyuan Liu, Maosong Sun, Bo Zhang, Leyu Lin. A Unified Framework for Community Detection and Network Representation Learning. IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), 2018. **(CCF A)**.

[5] **Cunchao Tu**, Zhiyuan Liu, Maosong Sun. Tag Correspondence Model for User Tag Suggestion. Journal of Computer Science and Technology (JCST), 2015. **(CCF B)**

[6] **Cunchao Tu**, Zhiyuan Liu, Huanbo Luan, Maosong Sun. PRISM: Profession Identification in Social Media. ACM Transactions on Intelligent Systems and Technology (ACM TIST), 2017. **(SCI)**

[7] **Cunchao Tu**, Zhiyuan Liu, Huanbo Luan, Maosong Sun. PRISM: Profession Identification in Social Media with Personal Information and Community Structure. National Conference of Social Media Processing (SMP'15), 2015. **(EI)**

[8] **Cunchao Tu**, Zhiyuan Liu, Maosong Sun. Inferring Correspondences from Multiple Sources for Microblog User Tags. National Conference of Social Media

Processing (SMP'14), 2014. **Best paper award**. **(EI)**

[9] Zhiyuan Liu, **Cunchao Tu**, Maosong Sun. Tag Dispatch Model with Social Network Regularization for Microblog User Tag Suggestion. International Conference on Computational Linguistics (COLING'12), 2012. **(CCF B)**

[10] Huimin Chen, Maosong Sun, **Cunchao Tu**, Yankai Lin, Zhiyuan Liu. Neural Sentiment Classification with User and Product Attention. The Conference on Empirical Methods in Natural Language (EMNLP'16), 2016. **(CCF B)**

[11] Ayana, Shiqi Shen, Yankai Lin, **Cunchao Tu**, Zhiyuan Liu, Maosong Sun. Recent Advances on Neural Headline Generation. Journal of Computer Science and Technology (JCST), 2017. **(CCF B)**

[12] Cheng Yang, Zhiyuan Liu, Maosong Sun, **Cunchao Tu**. Fast Network Embedding Enhancement via High Order Proximity Approximation. International Joint Conference on Artificial Intelligence (IJCAI'17), 2017. **(CCF A)**

[13] Xiangkai Zeng, Cheng Yang, **Cunchao Tu**, Zhiyuan Liu, Maosong Sun. Chinese LIWC Lexicon Expansion via Hierarchical Classification of Word Embeddings with Sememe Attention. AAAI Conference on Artificial Intelligence (AAAI'18), 2018. **(CCF A)**

[14] Zikun Hu, Xiang Li, **Cunchao Tu**, Zhiyuan Liu, Maosong Sun. Few-Shot Charge Prediction with Discriminative Legal Attributes. International Conference on Computational Linguistics (COLING'18), 2018. **(CCF B)**

[15] **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.

[16] **涂存超**, 刘知远, 孙茂松. 社会媒体用户标签的分析与推荐. 图书情报工作, 2013, 57 (23), 24-30, 35.

[17] 刘知远, 张乐, **涂存超**, 孙茂松. 中文社交媒体谣言统计语义分析. 中国科学: 信息科学, 2015, 12:1536-1546.