北京大学

博士研究生学位论文

题目：大规模图的解构及其在图挖掘任务中的应用

姓　　名：	吕天舒
学　　号：	1501111361
院　　系：	信息科学技术学院
专　　业：	计算机科学与技术（智能科学与技术）
研究方向：	数据挖掘与知识发现
导师姓名：	张岩教授

二〇二〇年六月

版权声明

摘要

我们处在一个互联的世界。在现实社会和数字生活之中，我们所面对的世界能够以图（graph①）为数据结构来刻画。图是一种适于刻画关系的数据结构。图无处不在：人群交流、社交、搜索、消费等种种的行为在图中发生，图亦在无形中影响参与者的决策。

线上社交网络、移动支付的出现加强了人与互联网之间关联，所产生的图规模巨大，信息多源。丰富的数据和应用场景令许多数据挖掘、机器学习的研究者们将目光投向了大规模图的图挖掘领域。图挖掘的经典任务，例如聚类、异常点挖掘、边预测等等，皆可视作在图的不同粒度上（节点、边、子图）进行的数据挖掘、机器学习任务[1]。节点标签分类任务[2]可以应用于用户偏好标签预测、高影响力用户识别等；边预测任务[3]可以用于推荐系统、知识图谱的构建等；子图分类[4]可以用于用户群体的划分、群体结构的演化预测等。由于图与现代人类社会的关系极为密切，大规模图的图挖掘具有重要的学术、应用和社会意义。

本文主要关注大规模图挖掘任务中普遍存在的三个挑战，并针对挑战提出了图解构的策略，设计了以解构为核心的图挖掘算法，展示了多个应用场景。具体的创新点展示如下：

- 本文对于大规模图挖掘领域中的重要共性问题进行提炼与总结。挑战一：同质性形式多样，节点的相似可以定义在很多个维度上。挑战二：图结构信息缺失，边、点、以及各类属性信息都有可能缺失。挑战三：临近度计算复杂，算法复杂度是限制其在大规模图中应用的主要因素。
- 本文提出图挖掘算法设计中的思想 — 图的解构策略，以应对提出的三大挑战。图的解构，顾名思义，指的是将图分解为直径较小的子图。本文中进一步列举图解构的两种实现形式：邻域解构和路径解构。邻域解构指的是分析单独节点时，将节点所处的局部子图截取出来加以分析。该解构方式可以应对同质性形式多样和图结构信息缺失的挑战。路径解构指的是分析图中节点之间的关系时，通过随机游走的方式截取节点对之间的短路径。该解构方式可以应对图结构信息缺失和临近度计算复杂的挑战。
- 本文基于邻域解构的策略，提出了一个大规模图上的重叠社区发现算法 — Fox，并展示了社区发现任务的应用方向：识别高影响力节点。Fox 算法的核心步骤为每个节点依次考虑其与邻居节点的关系，从所处的局部子图中选择一个最合

① 文献中，network 和 graph 常常混用。

适的邻居节点共同加入一个社区。通过将邻域解构引入算法，每个节点可以自动选择结构相关性更强的邻居节点（应对同质性挑战），社区自动地被划分出来（应对结构缺失挑战）。作为一种重要的结构特征，社区发现与图中的信息传播关系密切，可以辅助节点影响力的建模。

- 本文基于路径解构的策略，提出了一个组合式图神经网络 — CNE，并展示了其应用场景：提出一个新的节点中心度指标。CNE 算法的目标是将图中的节点映射到高维隐空间中，在隐空间中节点之间的距离可以反映其在原始图上的临近程度。通过将路径解构引入算法，节点之间的关系被随机游走短路径捕捉到（应对临近度计算复杂挑战），同时少数边、节点的缺失也不会影响算法整体的效果（应对结构缺失挑战）。随机游走的灵活性和高效性使得捕捉节点之间的关系更加容易，更加全面。
- 本文基于邻域解构和路径解构的策略，提出了一个考虑结构相似的图表示学习算法 — SNS，并展示了该图表示学习方法的应用方向：识别跨结构洞节点。SNS 算法的目标是在隐空间中保留节点之间的两种相似度：临近度和结构相似度。其中前者利用路径解构的策略实现，后者利用邻域解构的策略实现。基于两种不同的解构方式，表示向量可以保留更丰富的图结构信息，使得诸如跨结构洞节点的识别任务效果得到提升。

关键词：图的解构，图挖掘，社区发现，图表示学习

Large-scale Graph Decomposition and Its Application

Tianshu Lyu (Key Lab of Machine Intelligence, EECS)
Directed by Prof. Yan Zhang

ABSTRACT

People live in a connected world. In the physical and digital life, a network (graph) of entities and relations could be made in order to depict the mutual connections among the variables and processes in the world. Graph is a powerful and versatile data structure that easily allow us to represent real life relationships between different types of data (nodes). Graphs are everywhere. Behaviors in our daily life, including communication, social activity, information retrieval, transaction, happen in the graph. The graph topology also influences the behaviors of every participant.

Recently, graph mining techniques spurred attention from the data mining researchers for the soaring popularity of online social network and mobile payment. Typical graph mining tasks, including clustering, anomaly detection, link prediction, and so on, could be regarded as the classification task of different levels (node, edge, subgraph). Node label classification is able to predict user preferences, detect influential users; link classification is able to work as a recommendation system, build knowledge graph; subgraph classification divides user communities and predict the evolving trends of the communities. Because of the close connections between graphs and daily life, graph mining research has significance in academia, industry, as well as practical life.

The thesis focused on three ubiquitous challenges in graph mining tasks. Aiming at these challenges, the thesis proposed network decompostion, designed effective graph mining methods based on network decomposition, presented the performance in several applications. The specific contents is as follows:

- The thesis summarized the challenges in graph mining tasks. Challenge 1: Homogeneity is presented in various forms. Challenge 2: Network structure is not able to be collected completely. Challenge 3: The calculation of node proximity is complex.
- The thesis proposed network decomposition as the design principle of graph mining methods, in order to deal with the above challenges. Network decomposition is to divide an arbitrary graph into small-diameter connected components. In the thesis, the

network decomposition is further realized by neighborhood decomposition and path decomposition. Neighborhood decomposition refers to focus on the target node and its k-step neighbor induced subgraph when analyzing the target node. Path decomposition refers to focus on the truncated random walk path between the target pair of nodes when analyzing the relationships between the target pair of nodes.

- In Chapter 4, based on the neighborhood decomposition principle, an overlapping community detection algorithm of large-scale network, Fox, is proposed. Applying Fox to detect influential nodes is also presented. Fox let every node assess its relation with the neighbors in turn, choosing the closest neighbor from the ego-network to stay in the same community. By introducing neighborhood decomposition, every node automatically choose the closest neighbor (handle Challenge 1), communities are also automatically detected (handle Challenge 2). Community structure is a very important topology feature, which is further used in the influence maximization problem by incorporating with information diffusion models.
- In Chapter 5, based on the path decomposition principle, a compositional graph neural network, CNE, is proposed. A new node centrality is also presented. CNE aims at learning the latent representations of nodes so that node relationships could be inferred by the vector distances. By introducing path decomposition, node relationships are captured by short random walks (handle Challenge 3). Meanwhile, missing few nodes or edges can hardly influence the final results (handle Challenge 2). The flexibility and efficiency of random walks make it easier to capture node proximities.
- In Chapter 6, based on two types of decompisition principle, a graph representation method, SNS, is proposed. The detection of structural hole spanners is presented as an application. SNS tries to capture two types of node similarities, geodesic distance and structural similarities. The former is captured by path decomposition principle, while the latter is captured by neighborhood decomposition principle. By incorporating two decompositions, the learnt representation is able to preserve more network information, improving the performances of some topology-relevant tasks.

KEYWORDS: Network Decomposition, Graph Mining, Community Detection, Network Representation

目录

第一章 引言 **1**
1.1 研究背景 1
1.2 研究问题 2
1.2.1 解构的定义 3
1.2.2 关键问题 3
1.3 研究内容 4
1.4 论文组织结构 6

第二章 相关研究 **9**
2.1 图表示方法 9
2.1.1 社区发现 9
2.1.2 节点嵌入表示 11
2.1.3 图卷积网络 13
2.2 图解构策略 17
2.2.1 路径解构 18
2.2.2 邻域解构 20
2.3 本章小结 21

第三章 图挖掘任务中的解构策略 **23**
3.1 图的解构 23
3.1.1 节点属性、导出子图与子图解构 23
3.1.2 节点对关系、随机游走与路径解构 24
3.2 解构策略的应用 24
3.2.1 图挖掘任务中的三大挑战 24
3.2.2 解构策略有效应对挑战 25
3.3 本章小结 27

第四章 图挖掘中邻域解构策略的运用 **29**
4.1 大规模重叠社区发现算法 Fox 29
4.1.1 研究背景 29
4.1.2 模型描述 30
4.1.3 快速算法的合理性 37

4.1.4 算法复杂度 37
4.1.5 带权图上的 Fox 算法 37
4.1.6 实验评测 38
4.2 识别高影响力节点 46
4.2.1 应用背景 46
4.2.2 节点的权威性和差异性定义 48
4.2.3 权威性和差异性的平衡 49
4.2.4 差异性影响力最大化算法 50
4.2.5 实验测评 53
4.3 本章小结 55

第五章 图挖掘中路径解构策略的运用 **57**
5.1 基于路径解构的图神经网络 CNE 57
5.1.1 研究背景 57
5.1.2 模型描述 59
5.1.3 模型变种 61
5.1.4 实验评测 62
5.2 一个基于随机游走的节点中心度指标 68
5.2.1 应用背景 68
5.2.2 指标的形式化表示 69
5.2.3 新指标、子图中心度与 PageRank 70
5.2.4 利用局部子图近似计算中心度 71
5.2.5 新指标与图表示学习的关系 72
5.2.6 新指标与已有中心度的对比 74
5.2.7 实验评测 75
5.3 本章小结 80

第六章 图挖掘中路径解构策略和邻域解构策略的综合运用 **81**
6.1 节点结构信息辅助的图表示学习 SNS 81
6.1.1 研究背景 81
6.1.2 随机游走模型的局限性 83
6.1.3 利用特殊子图刻画节点局部结构 84
6.1.4 模型描述 88
6.1.5 实验评测 91
6.2 识别跨结构洞节点 98

6.2.1 应用背景 98
6.2.2 基于随机游走的节点结构向量 RWSig 99
6.2.3 RWSig 与图谱的关系 101
6.2.4 实验测评 104
6.3 本章小结 108

第七章 总结与展望 **111**
7.1 本文工作及创新性 111
7.2 未来工作展望 112

参考文献 **115**

个人简历、在学期间的研究成果 **127**
.1 个人简历 127
.2 发表的学术论文 127
.3 参与编写的书籍 128
.4 参与的科研项目 128
.5 所获奖励 128

致谢 **129**

北京大学学位论文原创性声明和使用授权说明 **131**

插图

1.1 国际金融网络。节点代表金融机构，有向带权边代表之间的关系。节点颜色代表机构所处的地理位置（大洲）。 2
1.2 YAHOO 问答网络示意图。这是一个异构图，包含用户节点和帖子节点。基于图结构和文本信息等，可以对用户兴趣、帖子领域进行分类，进一步挖掘其中的热门帖子和意见领袖。 2
1.3 本文研究思路及内容安排。 6

2.1 重叠社区概念。 10
2.2 Zachary Karate Club 社交网络以及它对应的二维映射空间。相同颜色的节点代表它们所属的社区。二维映射是通过 DeepWalk 算法计算得到的。 11
2.3 传统卷积操作和图上的卷积操作。 15
2.4 神经网络中的解构策略举例：Mixture-of-Experts 框架[63]。 18
2.5 邻居聚合计算框架[1]。 21

4.1 非连通社区划分结果的产生。(a) 和 (b) 是 k^{th} 轮迭代和 $(k+1)^{th}$ 轮迭代社区划分的快照。 36
4.2 在带社区标签数据集上的 NMI 值。 42
4.3 在带社区标签数据集上的 F1 值。 42
4.4 在通话图上应用 Fox 时，随着迭代次数的上升，节点的不同种类移动次数极速下降。 45
4.5 在通话图上应用 Fox 时，随着迭代次数的上升，$\widehat{\mathrm{WCC}(P)}$ 的值迅速达到稳定状态。 45
4.6 影响力最大化的传统方法所不适用的情形。 47
4.7 三种效用方程曲线。 50
4.8 在 Epinions 数据集中，不同融合方式对最终影响力的影响。 54
4.9 Imdb 数据集结果的差异性测评。 55

5.1 组合式图神经网络框架。对于一个正样本：目标节点 v 和其邻居 u，算法随机采样 K 个非邻居节点 $\bar{u}$。优化的目标是根据节点的表示向量将正样本和负样本区分开，其中表示向量是由节点的属性融合得到。 59
5.2 下一个点击的商品在相关性排序序列中位置的分布。 66

5.3 被不同节点中心度标记的足球图数据集。 74
5.4 节点所属的社区数目 (Amazon 数据集) 与节点的不同中心度。 77
5.5 节点所属的社区数目 (DBLP 数据集) 与节点的不同中心度。 78
5.6 新边生成的优先连接机制。 79

6.1 在图 (a) 和 (b) 中, 两个被虚线连接的节点代表两点之间有很多中间节点。节点颜色代表它们所属的组。(a) 和 (b) 展示了划分节点的两个角度。在 (a) 中，相互连接紧密的节点被划分为一组。在 (b) 中，相同地位的节点被划为一类。(c) 展示的是节点 0,2,8 的特殊子图度向量 (GDV)。此处只展示了向量的前 14 维。$E(\cdot,\cdot)$ 是两个向量的欧式距离。 83
6.2 在两种情况下，i 和 j 出现在同一窗口的概率较大。一是二者之间是直接连接，二是二者之间是强连接。 85
6.3 GDV 中涉及到的小子图[171]。相同颜色的节点属于同一个轨道。 85
6.4 CBOW 算法框架。$w_1, w_2, \cdots, w_W$ 是目标词的上下文。词表的大小是 V。窗口大小是 W。$\boldsymbol{W}$ 是输入层和隐层之间的权重矩阵。$\boldsymbol{W}$ 的每一行是一个 N 维的词向量 $\boldsymbol{v}_w$。类似的，$\boldsymbol{v}'_w$ 是 $\boldsymbol{W}'$ 中的行向量。 88
6.5 将结构信息与图表示学习的过程相结合。这里只展示单词 w_n 的学习过程。 89
6.6 Les Miserables 角色共同出现网络。节点的布局利用原子引力斥力法: ForceAtlas2[174]。 92
6.7 利用 DeepWalk[10] 来学习 Les Miserables 图的节点表示 ($p = 1, q = 1, d = 16$)。节点向量通过 PCA 算法降维到二维空间。 92
6.8 利用 node2vec[66] 来学习 Les Miserables 图的节点表示 ($p = 1, q = 2, d = 16$)。节点向量通过 PCA 算法降维到二维空间。 93
6.9 利用 Sns 来学习 Les Miserables 图的节点表示。节点向量通过 PCA 算法降维到二维空间。 93
6.10 数据中训练集的比例变化时，算法结果的 Macro-F1 分数和 Micro-F1 分数。 97
6.11 BlogCatalog 数据集上 Sns 算法的参数敏感度。 98
6.12 一张链状图的拉普拉斯矩阵特征向量可视化。图中绘制第 2^{nd}, 3^{rd}, 4^{th},14^{th}, 15^{th} and 16^{th} 个拉普拉斯矩阵特征向量。 102
6.13 一张普通图的拉普拉斯矩阵特征向量可视化。图中绘制第 2^{nd}, 3^{rd}, 4^{th},14^{th}, 15^{th} and 16^{th} 个拉普拉斯矩阵特征向量。 103

6.14 利用几个节点结构向量表示学习方法来处理 (a) 杠铃网络和 (e) 缺少了 (1,4) 边的杠铃网络。利用 PCA 降维[183]的方法，学习到的节点表示向量被影射到二维空间。 106
6.15 通过可视化混淆矩阵，进一步分析分类器在各个类别上的表现差异。 . . 108

表格

2.1 图神经网络的任务需求及模型要求。 14

4.1 数据集的基本信息。 39
4.2 基线算法的基本信息。 40
4.3 Fox 提升离散图发现算法的结果。 43
4.4 通话图中社区检测结果的基本信息。 43
4.5 通话图社区检测结果评价。 44
4.6 Google+ 社区检测结果基本信息及评价。 44
4.7 不同迭代停止条件下，Fox 在通话图上的表现。未忽略边权 46
4.8 不同迭代停止条件下，Fox 在通话图上的表现。忽略边权 47

5.1 不同测试集大小的召回值 (任务 1)。 63
5.2 共同浏览网络中边预测任务的 Precision@k (任务 1)。 64
5.3 共同浏览网络中冷启动节点边预测的 Precision@k (任务 2)。 64
5.4 浏览后购买网络中边预测任务的 Precision@k (任务 3)。 65
5.5 用户的点击序列以及各个算法给出的排在前列的相关商品 [1]。 67
5.6 Node Conductance 和已有中心度的排序的相关性。 73
5.7 静态图数据集的基本信息。 75
5.8 Flickr 数据集的四个快照。 75
5.9 全局节点中心度的运行时间（秒)。 76
5.10 各个节点中心度的 Spearman 排序相关系数 ρ[7]。 76

6.1 BlogCatalog 数据集的节点分类任务中，不同轨道的相对重要性 RI_o。 . 87
6.2 图中边保持不变的比例变化时，节点结构特征向量的变化。 107
6.3 跨结构洞节点的识别。表中展示的是训练比例为 1% 到 9% 时的 micro 和 macro F1 值。 107

第一章　引言

北岛《太阳城札记》组诗中的收尾一字诗[5]：

《生活》
网。

1.1　研究背景

回顾历史，网络科学原先是图论 (Graph Theory) 中的一个领域，所研究的对象是由若干给定的点及连接两点的线所构成的图形。自从 1736 年数学家欧拉利用图论解决了“七桥问题”，图以及图论开始被用来解决生活中的各种问题，图论也被证明是最有用的数学工具之一。典型的图论问题包括图中的最大流量问题，最大效能的工作分配问题，以及地图填色问题等等。20 世纪，图被应用在了诸如社会科学和经济学等等的更多的领域，并且取得了重大的进展。Erdös 和 Rényi 将概率论引入图论之中，并开创了一个全新的子领域---随机图论。概率论使得图论在组合优化的问题中有了更强的表现。

然而，网络科学在接下来的发展之中更多地关注真实的关系型数据。实际上，现实生活中的图与 Erdös 和 Rényi 定义的随机网络大相径庭。在近 20 年中，针对复杂网络的研究逐渐兴起。复杂网络具有高度不规则的结构，动态的演变，以及局部和整体特征不一致等等特点。这部分的代表性工作包括 Watts 和 Strogatz 提出的小世界模型 (Small-world Networks)；Albert 和 Barabási 提出的无标度网络 (Scale-free Networks)、优先连接理论 (Preferential Attachment) 等等。

随着互联网的爆发，网页链接网络，线上社交网络等一系列大规模图数据不断产生，这些数据中不仅实体相互关联组成图结构，实体和关系本身还具备丰富的属性信息，数据规模远远大于传统研究中研究对象的规模。机器学习和数据挖掘的研究人员意识到：对于这种类型的数据，学术界缺少有力的分析工具。由此，网络科学研究与数据挖掘结合紧密，新的研究方向 — 图挖掘应运而生，众多相关的研究问题和算法被提出。更重要的是，多个应用领域也意识到了图挖掘的重要性，进一步拓宽了图挖掘技术应用的场景。举例来说，在线上论坛上，用户之间就不同话题进行发帖、回帖等操作，形成一张用户行为图。根据用户间的行为关系、帖子文本内容，社区发现技术可以将用户划分为多个兴趣群组，传播行为分析可以发现群体中的意见领袖，图表示学习技术可以支持回答者推荐系统的搭建。金融领域中，国际机构、跨国公司之间形成包括借贷关系、股权所属、交易等复杂的金融连结。对该图或者子图的分析可以有

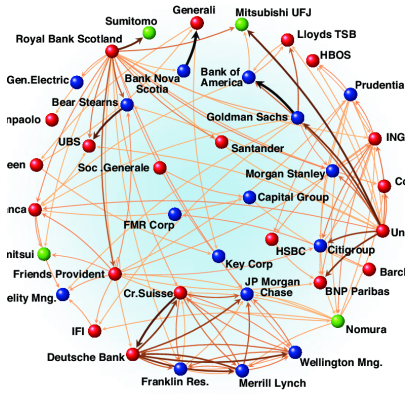


图 1.1　国际金融网络。节点代表金融机构，有向带权边代表之间的关系。节点颜色代表机构所处的地理位置（大洲）。

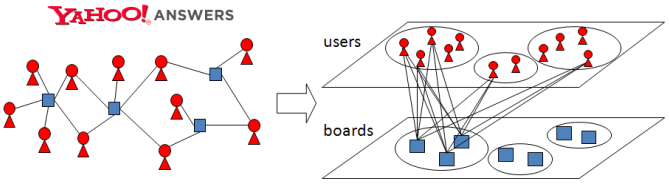


图 1.2　YAHOO 问答网络示意图。这是一个异构图，包含用户节点和帖子节点。基于图结构和文本信息等，可以对用户兴趣、帖子领域进行分类，进一步挖掘其中的热门帖子和意见领袖。

效地预警金融风险、识别潜在的违法行为[6]。

本文并不局限于某一或某几个应用场景，而是从方法论角度，研究图的解构策略在大规模图的图挖掘领域中的作用，对该领域的算法设计和算法应用都有重要的现实意义。

众所周知，解构是工程中常规的策略。例如分治算法，将一个规模为 N 的问题分解为 K 个规模较小的子问题，这些子问题相互独立且与原问题性质相同。求出子问题的解，就可得到原问题的解。通过分治，化繁为简，原问题的算法复杂度大大降低。再如模块化设计，在进行程序设计时将一个大程序按照功能划分为若干小程序模块，每个小程序模块完成一个确定的功能，并在这些模块之间建立必要的联系，通过模块的互相协作完成整个功能的程序设计方法。通过模块化设计，组合泛化，小程序模块得以大规模复用。针对不同的任务，重新设计模块之间的联系即可。

解构策略与图挖掘任务有着自然的契合之处。基本上所有图挖掘任务的基本假设是：待挖掘的知识蕴含于实体（节点）之间的关联（边）和组合（子图）之中。图挖掘算法挖掘出的模式（pattern）通常是图局部的模式，而非全局的模式。倘若追求保留全局信息，结果通常过于抽象、模糊，效用过低。因此图挖掘算法倾向于关注图的局部结构，建立局部的模型。当图规模巨大时，局部方法的另一个优势得到凸显：它们可以并行地处理大规模图，从而加速整个运算过程。总而言之，图解构策略在算法效果和计算效率上都能带来提升。

1.2　研究问题

本文涉及到解构策略以及其在图挖掘领域中的应用。本部分首先介绍解构策略的含义，再提出几个关键的研究问题。

1.2.1　解构的定义

分治是基本的算法设计思想[7,8]。

定义 1 (分治) 设 P 是待求解的问题。将 P 归约为 k 个彼此独立的子问题 $P_1, P_2, \cdots, P_k$。然后依此递归地求解这些子问题，得到解 $y_1, y_2, \cdots, y_k$。最后将这 k 个解归并得到原问题的解。

在数据挖掘尤其是图挖掘领域，鲜有工作提到分治思想。受到分治思想的启发，本文总结已有的图挖掘经典算法中与分治思想相似的设计方法，并将其称为图的解构策略。命名为解构的原因有两条：

1. 本文所涉及的图挖掘领域的算法都体现了问题的拆解过程。然而求解问题时，由于与具体的机器学习算法相关，分治思想所对应的递归求解子问题，归并子问题解往往不能在图挖掘算法中实现。因此，称之为图的解构更为准确。
2. 本文所涉及的图挖掘算法的核心计算通常围绕着图的拓扑结构。从算法设计的角度看，图上的分治思想将原问题分解为多个子问题。而从数据的角度看，每个子问题只关注原始图的子图。因此，称之为图的解构更为形象。

本文从子图结构的角度来划分图挖掘中的解构方式：

定义 2 (路径解构) 设 P 是待求解的图挖掘问题。将 P 划分为 k 个彼此独立的子问题 $P_1, P_2, \cdots, P_k$，每个子问题围绕着的原图中的一条**游走路径**进行计算。整理子问题的解 $y_1, y_2, \cdots, y_k$ 可以得到原问题的解。

定义 3 (邻域解构) 设 P 是待求解的图挖掘问题。将 P 划分为 k 个彼此独立的子问题 $P_1, P_2, \cdots, P_k$，每个子问题围绕着原图中包含目标节点的**导出子图**进行计算。整理子问题的解 $y_1, y_2, \cdots, y_k$ 可以得到原问题的解。

1.2.2　关键问题

图挖掘任务可以大致分为三类：（1）节点分类[2]，（2）边预测[3]，（3）子图划分[4]。近十几年中，有大量针对三类任务的研究工作。节点分类任务的方法可以分为两类：利用随机游走的传播节点标签[9]，抽取节点属性并训练分类器进行判别[10]。边预测的方法包括定义节点相似度[11]，极大似然模型等[12]。子图划分的方法包括优化某个连接紧密度指标[13]，基于节点属性分类[14]。

总得来说，图挖掘模型或者直接基于图原始的邻接矩阵进行计算，或者先学习图表示向量，将图映射到低维向量空间，再利用机器学习模型完成具体的应用任务。第一类模型较为传统，对于邻接矩阵的操作直观、可解释性强，但是操作空间较为有限。第二类模型近年比较流行，低维向量空间能够整合更复杂多样的关系型数据，与机器

学习技术的结合也可以进一步提升任务的效果。本文重点关注第二类模型，试图较为全面地展示解构策略在图挖掘任务中的体现，因此在接下来的各章中皆包含图表示方法和应用两部分内容，分别展示两种解决路径中的解构策略。

基于上面对于图的解构策略的定义，本文提炼了与图的解构相关的三个关键研究问题。这三个关键研究问题会贯穿接下来章节中对于各个图表示算法和应用的讨论。针对这些问题的思考可以加深解构策略必要性的认识。

1. **邻域解构和路径解构各自适用哪些任务？**不同类型的解构方式有各自的适用范围和特长。进一步讲，在邻域解构和路径解构都不适用的领域，是否有其他形式的解构方式？在图挖掘领域以外，在其他领域是否可以借鉴解构策略？回答该问题有助于扩大解构策略的应用范围，改进已有算法、提出创新解法。
2. **利用解构策略是否对算法的精确度、效率等方面产生影响？**从本质上讲，解构策略将算法的计算限制在图的某个局部，仅仅利用图的一个子图。那么被算法所丢弃的信息是否会对算法结果造成影响呢？从计算效率的角度来看，解构策略是否会潜在地引入重复的计算？回答该问题有助于解构策略的合理利用。
3. **解构策略产生效果的根本原因是什么？**解构策略对图的拆解是直观的，对于算法效率的提升也是显然的。同时，对于解构策略有效性的深层次探究也是必需的。回答这个问题有可能涉及到社会心理学的研究、数学的论证等等，有一定挑战。在不同应用场景下，解构策略产生效果的根本原因也有可能不尽相同。与前两个问题相比，该问题更具深度。回答该问题亦可以加深对前两个问题的理解。

1.3 研究内容

本研究以图挖掘与解构策略为关键词展开。

- 图挖掘所关注的研究对象不仅仅是图论中的图（即由若干给定的点及连接两点的线所构成的图形），还包括广义上的关系型数据。这类数据被以图的形式来表示及展示，描述某些事物之间的某种特定关系，用点代表事物，用连接两点的边表示相应两个事物间具有这种关系，同时如果事物和关系有附加的标注信息，那么则会以特征或标签的形式在数据中呈现。在本文所讨论的几个算法中，由于算法的应用场景各异，图的定义不尽相同。其中章节4.1、5.2、6.1、6.2的算法关注的是传统的由节点集合和边集合组成的图，章节4.2和5.1则可适用于带有节点属性的关系型数据。
- 解构是算法设计中重要的策略，然而，在数据挖掘领域中却鲜有关注解构策略的模型设计研究。本文从众多成功的图挖掘算法中提炼出广泛存在的解构策略，

并指出解构策略有能力应对上文提出的图挖掘领域中的重大挑战。在研究问题一节中，本文将解构策略划分为两种具体的实现方式：邻域解构和路径解构。其中邻域解构指的是：解构后的子问题围绕着原图中包含目标节点的**导出子图**进行计算。这种解构方式常见于节点属性的刻画与推断模型中。路径解构指的是：解构后的子问题围绕着原图中包含目标节点的**游走路径**进行计算。这种解构方式常见于节点之间关系的刻画与推断模型中。

受到神经网络、机器学习技术的快速发展的影响，近年，图挖掘研究与这些技术的结合日趋紧密。传统图挖掘算法直接处理图中的邻接关系、链路结构以完成任务。由于这种传统的解决方式与机器学习框架难以相融，以*表示学习 + 下游机器学习模型*为代表的新图挖掘算法框架逐渐成为研究的主流。表示学习将图结构、各类属性信息等融合加工，使得待处理的关系型数据转换为表示向量。下游模型以表示向量作为输入数据，借助丰富的机器学习工具完成各类图挖掘任务。有时算法亦会以“端到端”的方式实现，以关系型数据为输入，以具体的任务指标为输出。此时，表示学习和下游机器学习模型不是框架中显式的独立的两部分，而是隐式地融合，表示向量会以“隐层”的形式出现在机器学习模型中。但是总的来说，显式结合和隐式结合都符合*关系型数据 → 表示向量 → 任务相关模型*框架。

图的表示学习可以进一步分为离散表示和连续表示。社区发现的结果是一种典型的图的离散表示。对于离散社区发现结果来说，图中节点可以由一个长度为社区总数的 one-hot 向量表示，节点所属社区所对应位置上的元素为 1，其余位置元素为 0。重叠社区发现结果则可以对应一个 multi-hot 向量。图表示学习和图神经网络中的隐层是典型的图的连续表示。图中的每个节点映射到隐空间的一个位置，节点由位置向量刻画。

本文以解构策略为主线，介绍解构策略的功能和适用范围，展示图的表示学习方法与端到端图挖掘算法中解构策略的灵活运用。

邻域解构的运用：重叠社区发现任务的目标是将图中的节点分组，同组的节点相互连结紧密，属性更为相似。本文提出的算法 Fox，从模型功能上讲，将节点划分任务解构为一对相邻节点是否同组的二分类问题；从模型处理的数据上讲，将原始图解构为目标节点所处的导出子图。Fox 算法从节点邻域中挖掘出与中心节点更加相近的邻居，解决了同质性形式多样的问题；将原本不相邻的节点归为同一社区，缓解了图结构信息缺失的问题。在应用方面，本文基于社区发现的结果，定义节点集合影响力的差异性，快速识别高影响力高差异性的节点。

路径解构的运用：图神经网络的目标是对以图为形式组织在一起的复杂数据进行建模，从而支持节点标签预测、关系预测等任务。本文提出的算法 Cne，从模型功能上讲，将刻画图中一对节点的关系解构为计算随机游走路径中，两个节点依次出现的概率；从模型处理的数据上讲，将原始图解构为包含两个节点的游走路径。Cne 算法

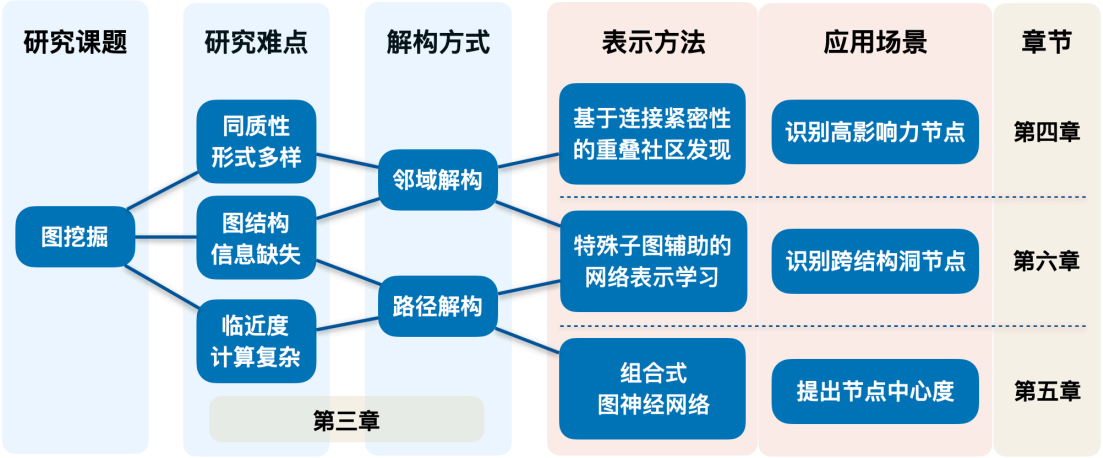


图 1.3　本文研究思路及内容安排。

综合多条游走路径，缓解了图结构信息缺失的问题；产生随机游走路径的代价远低于传统的临近度计算，解决了临近度计算复杂的问题。在应用方面，本文基于基于体现路径解构的图表示学习算法，提出一个新的节点中心度指标。指标的计算不再受限于计算复杂度过高，可以应用于大规模图。

邻域解构和路径解构的综合运用：图表示学习的目标是将图中的节点映射到隐空间中，使得隐空间中节点的相对位置可以反应节点在原图中的关系。本文提出的算法 SNS，既希望保留原图中节点对之间连结紧密的程度，又希望保留每个节点的地位（边缘或核心）。其中，利用路径解构实现刻画节点对关系的功能（同 CNE），利用邻域解构实现刻画节点局部地位的功能。在应用方面，本文将目标节点的邻域进行路径解构，提出一个高效的节点地位表示方法，以支持跨结构洞节点的识别。

图1.3展示了当前研究的整体路线图，其中两种解构方式分别对应三个研究挑战，基于两种解构方式分别提出了三种表示方法，以及两种解构方式在三个端到端任务中的应用。

1.4　论文组织结构

本文内容的组织安排如下：

- 第一章 **引言：**介绍了研究背景、研究问题和研究内容。
- 第二章 **相关研究：**按照“图的表示方法”和“解构策略”两条主线分别梳理相关文献。
- 第三章 **图挖掘任务中的解构策略：**本章从宏观角度论述了解构策略及其应用范围。
- 第四章 **图挖掘中邻域解构策略的运用：**本章提出了一个高效处理大规模图的重

叠社区发现算法，算法设计体现了邻域解构策略。随后，本章展示社区发现在识别高影响力节点的应用。本部分的工作发表在数据挖掘领域顶级会议 ICDM 2016。

- 第五章 **图挖掘中路径解构策略的运用**：本章提出了一个处理复杂节点属性的图神经网络，算法设计体现了路径解构策略。随后，本章提出了一个适用于大规模图的节点中心度指标。本部分的工作发表在推荐系统领域顶级会议 RecSys 2019 和数据挖掘领域会议 PAKDD 2020。
- 第六章 **图挖掘中路径解构与邻域解构策略的综合运用**：本章提出了一个同时考虑两种节点相似性的图表示学习算法，算法设计体现了邻域解构和路径解构策略。随后，本章提出一个高效的识别跨结构洞节点的算法。本部分的工作发表在信息检索领域顶级会议 CIKM 2017。
- 第七章 **总结与展望**：总结了本文的主要工作内容和研究贡献点，并且针对未来可能深入进行的研究给予了展望。

第二章　相关研究

在已有工作中，即便图的解构策略没有被正式提出，但仍有很多体现。本章将整理回顾一下近年来图挖掘领域中与解构策略相关的成果，包括数据挖掘领域重点关注的社区发现（Community Detection），节点嵌入式表示（Node Embedding Methods）和机器学习领域常用的图卷积网络（Graph Convolutional Networks）。鉴于本文提出了两种图解构的实现方式：邻域解构和路径解构，本章所整理的相关工作也将按照此种分类方式分类。本节将关注在数据挖掘和机器学习领域引起关注的几个重要算法，尤其是可以适用于大规模图（至少百万级节点）的算法以及与深度学习有交叉的一些算法。当然，对于图表示学习这一任务，还有很多相关工作：社交网络的隐空间学习[15]，流形学习[16]，几何深度学习[17]等。这些工作将不在本节中涉及，如果关注相关工作的话可以查阅[15–18]。

2.1　图表示方法

2.1.1　社区发现

直观的说，社区指的是图中的聚集的群体，社区内部的节点之间的联系相对紧密，但是各个社区之间的连接相对来说更加稀疏。其实，图中的社区现象在学术界已经是个比较古老的话题了，最早期的记录甚至来自于 80 年前。空手道俱乐部（Karate Club）[19]，科学家合作网络（Collaboration Network）和海豚群体（Dolphins）[20] 的社交行为研究是几个经典的社区结构的研究案例，其中著名的空手道俱乐部社区已经成为检验社区发现算法效果的标准（benchmark）之一。随着互联网和在线社交网站的兴起，在 Twitter，Facebook，Flickr 这样的用户生成内容（UCG）网站上使用社区发现的技术已经成为热潮。在这些社区中用户相互的交流与反馈，能为传统的社区带来丰富的内容信息和新的结构，提升用户的活跃度和使用黏性。

生活中的大部分图都有着重叠属性、层级属性，可以称之为重叠社区。图2.1展示的是一个最直观的实例。假设图中的黑色圆点代表一个人。每个人都有自己的朋友圈子、家人、校友、相同爱好的朋友以及科研圈子的伙伴，因此每个人都分属这五个圈子。而将研究方向这一群体再进一步分类，可以发现它内部又包含了四个圈子：数学、物理、生物、生物物理，这四个圈子也有相互交叉的人。另外，如果将家人圈子展开，可以进一步发现每位家人都会有属于他的家人圈子，这个圈子可以无限扩展下去。这就是重叠社区的概念，每个个体可以只属于一个社区，也可以属于多个社区结构之中。这

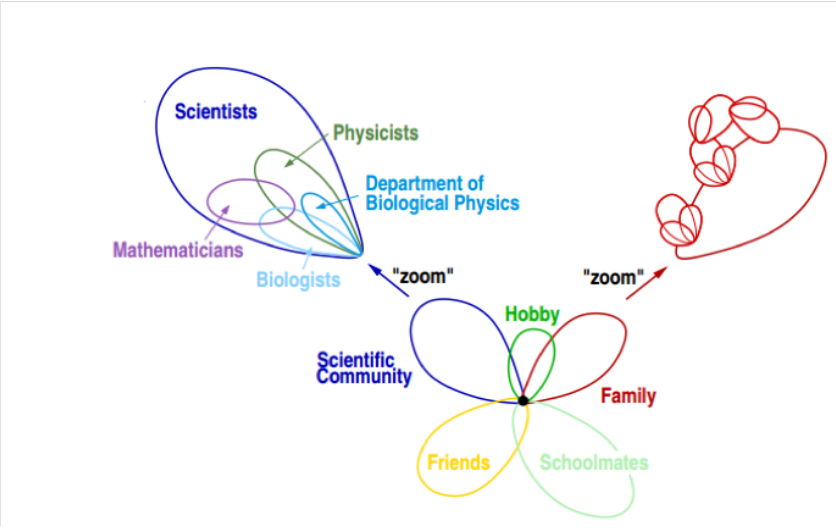


图 2.1　重叠社区概念。

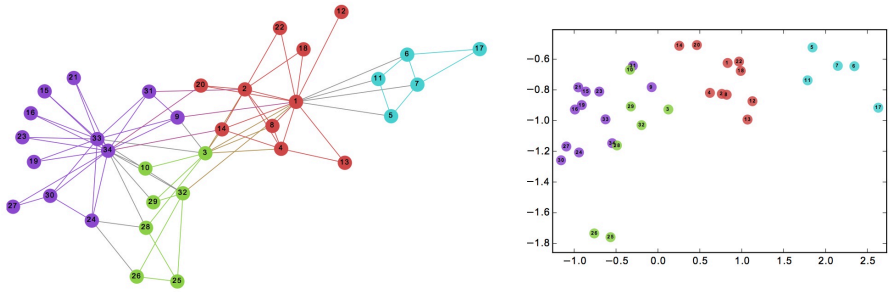
一概念具有普适意义，因为重叠社区广泛存在于社交网络、生物网络等等各种图中[21]。

- **非重叠社区发现。**非重叠社区发现的算法中，很大一部分是基于模块度[22] 的算法。模块度的定义是：

$$Q = \frac{1}{2m} \times \sigma_{i,j}[A_{ij} - \frac{k_i k_j}{2m}]\delta(C_i, C_j) \tag{2.1}$$

其中 k_i 代表 i 节点的度。当 i 和 j 节点一个社区中时，δ 函数等于 1，否则为 0。模块度定义是建立在社区之间无交叉部分的前提下的。当社区内部的连接密度大于图的平均水平时，该社区的模块度较大。为了实现模块度的最优化，有几种不同的算法策略：合成聚类的贪心算法（Agglomerative Greedy）[23] 或模拟退火法[24]。针对亿级节点的图，有一种多层级方法的解决方案[25]。但这些算法的一个巨大缺陷是：当图的规模增大时，检测的社区质量会随之下降[26]。而且，也有研究显示模块度是存在一些限制的。当图规模很大时，模块度无法检测出比较小的社区。一些非重叠社区算法是基于随机游走的。一般认为，在随机游走的过程中，因为社区内部的连接更加紧密，所以下一步停留在社区内部的概率会大于跨出社区的概率。Walktrap[27] 和 Infomap[28] 就是利用这种思想的两种算法。最终的社区结构是基于随机游走路径的。在对路径的存储过程中，需要采取一些压缩措施，以减少对存储空间的需要。在 Lancichinetti et al. 的研究中[26]，Infomap 是这类算法中的佼佼者。

- **重叠社区发现。**因为定义社区的角度不同，算法也有很多不同的种类。社团划分法（CPM[29], SCP[30]）对社区的定义最符合人们对社区的认识 — 完全子图，这类算法在连接程度高的图上比较适用。毕竟，图中很少会出现一个节点数大于 4 的完全连接子图。另一种发现社区的思路是从种子扩展（MNOC[31]，UEOC[32]，



(a) Input: Karate Graph　　(b) Output: Representation

图 2.2　Zachary Karate Club 社交网络以及它对应的二维映射空间。相同颜色的节点代表它们所属的社区。二维映射是通过 DeepWalk 算法计算得到的。

OSLOM[33]）。这种算法的设计思路是利用一个收益函数去衡量社区时候扩张，吸纳新的节点。因此，收益函数对算法的复杂度和划分结果的质量起了决定性的作用。这类算法的最差复杂度大多都达到了 $O(n^2)$。

也有一些算法借鉴了机器学习中的技术，用来完成社区发现的任务。P.K. Gopalan 和 D.M. Blei 提出的算法[34] 是基于混合隶属度随机快模型和变分法的。它利用一组隐变量来表示节点和社区之间的从属关系及从属的紧密程度。非负矩阵分解在机器学习中也是一个流行的技术（BigCLAM[35], CoDA[36]）。这类算法最大的瓶颈是巨大的时间、空间开销。

2.1.2　节点嵌入表示

表示学习又称作特征学习，是机器学习领域中的一个重要研究问题，它的目标是自动学习一个从原始输入数据到新的特征表示的变换，使得新的特征表示可以有效应用在各种机器学习任务中，从而把人从繁琐的特征工程中解放出来。图表示学习即是对图信息进行的表示学习，目标是将节点在图中的位置信息和邻接信息通过一个一个低维的向量表示出来。图2.2展示的是将 Zachary Karate Club 社交网络映射到二维空间的一个例子。可以看到在二维空间中，那些在原图中同在一个社区的点（通过同一颜色来表示）相互间的距离也是很近的。文献[37] 中有对图表示学习的详细定义：给定图 $G = (VE)$，G 对应的节点特征矩阵是 X，对任意节点 $v \in V$，学习低维向量表示 $r_v R^k$，r_v 是一个稠密的实数向量，并且满足 k 远小于 $|V|$。把节点表示为向量的 3 种好处[38]：

1. 机器学习算法可以直接利用得到的节点向量表示作为输入，因此无需针对图数据重新设计新的机器学习算法。
2. 在节点表示的向量空间中，可以进行各种各样的运算。也就是说，图中节点的距离、相似度等需要定量计算而又不容易给出明确定义的概念，在向量空间中则可以直接进行各种运算。

3. 在大规模图数据中，节点之间的链接关系可能会非常复杂而不易观察，但是通过在低维向量空间中进行可视化分析展示，可以很直观地观察节点之间的关系。

下文将针对上述各种情况下的图表示学习进行介绍。直观上可以从两种角度对不同的算法进行分类：一种是参考表示学习的分类，分为有监督的图表示学习和无监督的图表示学习；另一种是根据输入数据的不同进行划分，比如按照图的方向性、是否是异构图等性质。然而这两种划分的角度并不合适。因为当前的图表示学习算法的主要区别在于算法类型，同一算法类型下的算法框架都是相似的。

- **邻接矩阵及谱方法。**对图最传统、最直观的表示方式是邻接矩阵，刻画节点与节点之间的连结关系，包括边的方向、边的权重。谱方法在传统的图表示学习中占有很重要的位置。比如 PCA、LLE、Laplacian Eigenmaps 等等降维算法，在机器学习或者模式识别课程中都会有介绍。它们将高维的邻接矩阵降维，可以将在图中相邻或连接紧密的点映射到低维空间，使它们在低维空间上的距离也相应的比较近。然而，当面对现实中的大规模图时，这种传统的技术常常显得无能为力。具体体现在
    1. 计算复杂度过高。计算图中节点之间的关系时，传统算法通常以迭代的方式实现，这使得算法的计算复杂度很高。例如，统计两点之间的的可行路径数的计算是一个典型的组合问题。
    2. 可并行能力弱。为了处理大规模数据，常规的解决方案是设计分布式的可并行算法。而关系型的数据及其对应的算法在并行化的设计上存在着巨大的挑战。难点在于，关系型数据的独特性，图中的每个节点都与其他节点之间存在着连结。倘若简单地将图划分为多个子图组成的分块，有大量的计算会发生在块间，块与块之间势必也会涉及高昂的通信成本。
    3. 无法适应机器学习算法。目前，机器学习技术被广泛的应用于多个领域，为不同问题提供了统一的解决范式。然而，传统图表示学习算法与机器学习算法之间存在着使用上的障碍。机器学习算法通常认为输入的数据，如节点，可以由向量空间一个独立的向量来表示。然而图中，如上一点所述，节点与其邻居相关联，获取独立的节点向量表示成为了一个问题。
- **保留结构信息的图表示学习。**关系型数据最关键的信息是图的结构信息。诸如发现图中的高影响力节点、社区划分等图挖掘任务可以在原始空间（即邻接矩阵）中完成，然而如上所述，存在诸多问题。由此出发，一个自然的想法是学习一个新的低维的表示空间，该空间完全基于图中节点的连接关系，相关的图挖掘任务可以在该空间中高效、有效地被完成。典型的图结构信息包括连边关系[39]，邻域结构[10]，高阶的邻接关系[40]，社区结构[41]。这些算法分别将图中的节点映射到相应的隐空间中，在隐空间中，具备上述关系的节点映射到相近的

位置。从而在下游的机器学习模型中，具备上述关系的节点会有着相似的模型输出。

还有一些方法致力于保留图中固有的结构特性，例如图的连边传递性（也称为三元闭包性质）[42] 和正负符号图中的平衡性[43,44]。这些固有特性在与图结构的演化与预测的问题上有重要作用。

- **保留附加信息的图表示学习。**除了图的结构信息，现实生活中的图数据通常还伴随着丰富的附加信息。以社交网络为例，每个用户节点对应一些其发布的博文，用户还有是否转发某条博文的标签，节点间的关注、点赞、评论关系对应不同标签的边，以及实际发生的时间戳[45]。这些附加信息为精确刻画节点之间的关系，完成特定任务同样作用巨大。尤其由于数据收集手段能力所限，图结构稀疏，大量连边没能正确捕捉，此时附加信息的存在变得尤为重要，它可以在一定程度上弥补结构信息的缺失。

  从方法上讲，这类图表示学习算法不得不面对的挑战是：如何合理掌控结构信息和附加信息在表示向量中的占比。最简单的处理方式直接拼接两部分表示向量[46]。也有一些工作与多模信息和多源信息融合技术相似[47,48]，将两部分用更智能的方式进行融合。

- **针对特殊任务的图表示学习。**在上面提及的类别中，大部分相关算法都是非监督的方法。模型以图的结构、属性、附加信息为输入，目标是希望学习到的向量空间尽可能地完整保存输入的信息。通常，这些蕴含丰富信息的表示向量会在下游任务模型中被当作输入数据，用以支持完成各种各样的任务。倘若以监督或半监督的方式来做图表示学习，可以直接利用质量更高的监督信息，端到端地学习更有针对性的向量空间。在计算机视觉[49] 和自然语言处理中[50]，端到端的模型都展示出突出的能力。

  在图表示学习中，设计端到端的模型同样可行。以图中的节点标签分类为例，模型可以以图的结构信息为输入，节点标签为监督信息，而节点的表示向量为模型的中间层。学习到的图表示向量是专门为节点标签任务服务的。类似的端到端图表示学习任务包括图信息传播任务[51]，异常检测[52]，图对齐[53]，学术合作预测[54]。

2.1.3　图卷积网络

2019 年伊始，阿里巴巴达摩院发布了年度十大科技趋势，其中之一是“超大规模图神经网络系统将赋予机器常识”。这已经不是图神经网络第一次被放到深度学习技术的头条位置了。2018 年，DeepMind、谷歌大脑、MIT 和爱丁堡大学的 27 名作者，对图神经网络及其推理能力进行了全面阐述[55]。随后，图卷积网络（Graph Convolutional Network,

表 2.1　图神经网络的任务需求及模型要求。

任务需求	模型需求
处理多种基于图的机器学习任务	产生节点、边、图的向量表示
处理任意大小和结构的图	模型参数与图的大小和结构无关
图中的节点编号是任意的	模型的输出与输入的节点顺序无关
利用图的结构和节点特征进行学习和预测	图上特征的传递/融合机制

GCN）、图神经网络（Graph Neural Network, GNN）、关系网络（Relation Network）、几何深度学习技术（Geometric Deep Learning）等关键词频频出现在各大顶级机器学习、数据挖掘会议上。

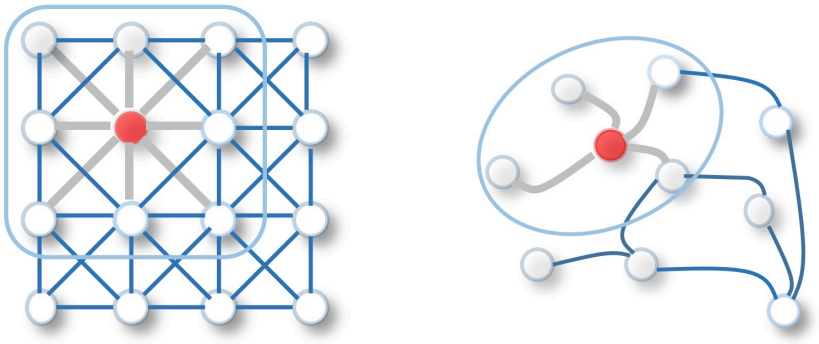
图神经网络（GNN）引起广泛关注的原因可以暂归为两点。首先，图作为一种更为常见的数据结构，目前却没有与之相对应的通用的神经网络模型。而 GNN 可以视作 CNN 在图（Graph）上的扩展，将卷积的思想从欧几里得域迁移到非欧几里得域[17]。第二个原因是，图神经网络具备的“推理能力”，而这恰恰是基于传统神经网络的人工智能系统所欠缺的能力。从一定程度上讲，图神经网络是符号主义（Symbolicism）和非符号主义（Non-symbolicism）相结合的产物（Neural-symbolic System），将规则、知识引入神经网络，使得神经网络具备了可解释性和推理能力。

图神经网络处理的对象是图（graph）。正如卷积神经网络可以处理任意大小的图片（image），循环神经网络可以处理任意长度的序列，通常对图神经网络也有一些期望，如表2.1所示。

基于该表可以对图神经网络有一个基本的认识。事实上，图神经网络并不是近一两年新诞生的技术，早在 2005 年这一概念即被提出。随后，计算机科学领域、理论物理复杂网络领域的研究者在图（graph）的空间域（Spatial Domain）和频谱域（Spectral Domain）上分别提出了不同形式的图神经网络，并最终在 2017 年实现了空间域模型和频谱域模型的融合。自此，深度学习技术和图神经网络迎来了广泛关注。图神经网络的核心在于：如何类比卷积神经网络在网格状数据上的卷积操作，来定义图上的卷积操作？进一步来说，卷积操作背后所对应的图片（image）的局部不变性（shift-invariance）和组合性（compositionality）是如何在图（graph）上体现的？

- 局部不变性：包含平移不变性、旋转不变性、尺度不变性。在卷积神经网络中，作用在局部区域的卷积核被整张图片所共享。
- 组合性：简单的卷积核所提取的基本特征可以组合成为复杂特征。在卷积神经网络中，随着网络层数的增加，网络可以逐渐探测到初级的边缘特征、简单的形状特征直至复杂的图像特征（如指纹、人脸）。

空间域模型选取目标节点的邻居进行卷积操作，更易于理解；频谱域模型以图信



(a) 2D Convolution. Analogous to a graph, each pixel in an image is taken as a node where neighbors are determined by the filter size. The 2D convolution takes the weighted average of pixel values of the red node along with its neighbors. The neighbors of a node are ordered and have a fixed size.

(b) Graph Convolution. To get a hidden representation of the red node, one simple solution of the graph convolutional operation is to take the average value of the node features of the red node along with its neighbors. Different from image data, the neighbors of a node are unordered and variable in size.

图 2.3　传统卷积操作和图上的卷积操作。

号处理（Graph Signal Processing）为基础，更具数学形式上的美感。不过，两类模型都对局部不变性和组合性给出了各自的阐述和实现方式，从而奠定了两类模型最终走向融合的基础。

- **空间域图卷积模型**。图卷积网络的目标是以一种端到端的方式解决与图相关的机器学习任务，它可以被视作针对图结构数据的深度学习技术。受到经典的卷积模型 CNN 和 RNN 以及深度学习中的自编码器技术（autoencoder）的启发，近年来一系列的相关概念扩展到了图数据之上。例如，在图上的卷积操作是由二维网格上的卷积操作扩展而来。在图2.3中，图像（image）可以被视作一种特殊的图，图像中的每个像素点与其相邻的像素点有边相连。那么以此类比，在图中的卷积操作可以通过邻居节点的加权求和的方式来实现。

  GAT[56] 和 GraphSAGE[57] 是典型的空间域图神经网络。通过展示这两个经典算法，可以对空间域的图卷积框架可见一斑。

  GAT 的核心操作是基于多头注意力机制（Multi-head Attention）的邻域卷积，具体的计算公式为：

$$\boldsymbol{y}_i = \sigma\left(\frac{1}{M}\sum_{m=1}^{M}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^{(m)}\left(\boldsymbol{x}_j\cdot\boldsymbol{W}^{(m)}\right)\right).$$

  其中，$\boldsymbol{x}_j\in\mathbb{R}^{1\times d_x}$ 是节点 j 的输入特征，$\boldsymbol{y}_i\in\mathbb{R}^{1\times d_y}$ 是节点 i 的输出特征，$\mathcal{N}_i$ 是节点 i 的邻居节点集合，M 是头（head）的个数；对于第 m 个头，$\alpha_{ij}^{(m)}$ 是该头中节点 i 对节点 j 的注意力系数，$\boldsymbol{W}^{(m)}$ 是对应的线性变换矩阵。

GraphSAGE 的核心操作包括两个步骤：1. 融合目标节点邻域的特征；2. 将邻域融合的特征和节点本身的特征进行拼接，通过神经网络更新每个节点的特征。具体公式如下：

$$\boldsymbol{h}_{\mathcal{N}_i} = \mathrm{aggregate}(\{\mathrm{x_j} \mid \mathrm{j} \in \mathcal{N}_\mathrm{i}\}); \mathrm{y_i} = \sigma\big(\mathrm{concat}(\mathrm{x_i}, \mathrm{h}_{\mathcal{N}_\mathrm{i}}) \cdot \mathrm{W}\big).$$

这里的 aggregate 操作有丰富的选择，如最大值池化（max-pooling）、平均值池化（average-pooling）、LSTM 等。上述步骤反复进行 K 次，每个节点就可以和它的 K 度邻居的特征做融合。

- **频谱域图卷积模型。**图谱是图的拉普拉斯矩阵的特征值。具体来说，给定图 $G(V, E)$，其中 V 和 E 分别是 G 的点集和边集，点集大小为 n，边集大小为 m。图 G 的邻接矩阵 $\boldsymbol{A}$ 为一个 $n \times n$ 的矩阵：A_{ij}=1 代表 i, j 节点之间有边相连，否则 A_{ij}=0。图的度矩阵为 $\boldsymbol{D}=\mathrm{diag}(\mathrm{d_1}, \mathrm{d_2}, \cdots, \mathrm{d_n})$，其中 d_i 是第 i 个节点在图中的度数（degree）。图的拉普拉斯矩阵定义为 $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{A}$，对其进行特征值分解：

$$\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top,$$

其中 $\boldsymbol{\Lambda}=\mathrm{diag}(\lambda_1, \lambda_2, \cdots, \lambda_\mathrm{n})$ 是按照特征值从小到大的顺序排列的，$\boldsymbol{U}=[\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_n]$ 是对应的特征向量组成的正交矩阵。上述特征值集合 $\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ 即为图 G 的图谱。注意，尽管 $\boldsymbol{A}$ 会随着 G 的节点编号的改变而改变，但其图谱却不会改变，它只与图的抽象结构有关。

频谱域图神经网络的发展主要是围绕着图上卷积核的设计来进行的。借鉴图卷积网络（GCN）模型论文[58] 的思路简要梳理这一发展过程。

根据卷积定理，两个函数卷积后的傅立叶变换，是它们各自函数傅立叶变换后乘积；也就是说，两个函数的卷积是它们各自傅立叶变换乘积的逆变换。由此可以给出图信号 $\boldsymbol{x}$ 与卷积核 $\boldsymbol{h}$ 在图 G 上的卷积形式。在上一问题中，为形象地展示拉普拉斯矩阵的含义，可以将 $\boldsymbol{x}$ 定义为长为 n、元素为实数的向量。在实际应用时，可以进一步将每个节点所对应的实数扩展为一个节点的属性向量，可以是节点的类型标签、节点的属性表示向量、节点的结构特征（度数、PageRank 值等）。$\boldsymbol{x}$ 和 $\boldsymbol{h}$ 的傅立叶变换为：

$$\hat{\boldsymbol{x}} = \boldsymbol{U}^\top \boldsymbol{x}, \qquad \hat{\boldsymbol{h}} = \boldsymbol{U}^\top \boldsymbol{h}.$$

则二者的卷积（即各自傅立叶变换乘积的逆傅立叶变换）为：

$$\boldsymbol{x} * \boldsymbol{h} = \boldsymbol{U} \cdot \mathrm{diag}\big(\hat{\mathrm{h}}\big) \cdot \mathrm{U}^\top \mathrm{x}.$$

初级的频谱域图神经网络直接将 $\mathrm{diag}(\hat{\mathrm{h}})$ 替换为 $\mathrm{diag}(\boldsymbol{\theta})$，其中 $\boldsymbol{\theta}=[\theta_1, \theta_2, \cdots, \theta_n]$

为待学习的参数。添加激活函数 $\sigma(\cdot)$ 后，最终的输出值为：

$$y = \sigma\left(\boldsymbol{U} \cdot \mathrm{diag}(\boldsymbol{\theta}) \cdot \mathrm{U}^{\top}\mathrm{x}\right) .$$

这个模型中 $\boldsymbol{U}\cdot\mathrm{diag}(\boldsymbol{\theta})\cdot\mathrm{U}^{\top}$ 部分的计算量为 $\mathrm{O}(\mathrm{n}^2)$，计算复杂度过高，对于大规模图很难计算。

进一步地，将上述模型中的 $\mathrm{diag}(\hat{\mathrm{h}})$ 替换为 $\sum_{k=0}^{K}\alpha_k\boldsymbol{\Lambda}^k$，其中系数 $\{\alpha_k\}_{k=0}^{K}$ 是待学习的参数。由此，得到无需特征分解、计算复杂度为 $\mathrm{O}(\mathrm{n})$ 的 GCN 模型：

$$\boldsymbol{y} = \sigma\left(\boldsymbol{U} \cdot \left(\sum_{k=0}^{K}\alpha_k\boldsymbol{\Lambda}^k\right) \cdot \boldsymbol{U}^{\top}\boldsymbol{x}\right) = \sigma\left(\sum_{k=0}^{K}\alpha_k\boldsymbol{L}^k\boldsymbol{x}\right) .$$

这种形式不仅大大简化了运算，还巧妙地具备了“局部性”。简单来说，$\boldsymbol{L}^k=(\boldsymbol{D}-\boldsymbol{A})^k$ 包含了 $\boldsymbol{D}^k$ 项（节点度数），$\boldsymbol{A}^k$ 项（节点的 k 度邻居个数），以及 $\boldsymbol{A}$ 和 $\boldsymbol{D}$ 的交叉项（不大于 k 度的邻居个数）。因此，图卷积网络（GCN）的卷积操作实际上是将每个节点的 K 度范围内的邻居的特征融合起来，这与空间域的图神经网络的做法不谋而合。

2.2　图解构策略

在数据挖掘领域，最大的挑战是设计出可以处理复杂的真实数据的算法。当问题本身很复杂时，分解问题是通常的有效解决方案。在这里，将复杂问题（或系统）分解为多个相互关联的子问题（或系统）的思想称为**解构**[59]。解构的目标是简化复杂问题的解决流程：首先将其分拆为多个可控制的子问题后，然后利用已有的工具一一有效解决子问题，最后将子问题的结果归结起来，原始的复杂问题得以解决。

尽管解构是一个简单有效的设计准则，在数据挖掘领域却鲜有研究直接关注这一策略，而大量的实践案例却在隐式地践行这一策略[60]。同时也有大量的研究与解构策略密切相关，比如分布式与并行计算[61]、集成学习（ensemble learning）[62]，其中有许多解构算法[59]。在神经网络中，亦有解构策略的体现。如图2.4所示，MoE（Mixture-of-Experts）框架将输入空间拆解，拆解相对自由，子空间之间允许出现重叠部分。每个专家网络分别计算一个拆解后的子空间，输出一个以当前子空间输入为条件的条件概率。一个门控网络用来控制多个专家网络输出结果的融合权重。权重本身并不是个常数，而同样也是与输入数据相关的变量。

解构操作的好处体现在多个方面：

- 提升分类效果。Sharkey[64] 认为解构的主要目的就是提升模型的效果。这个观点可以被平衡方差和偏差的理论（bias-variance tradeoff）来解释。由于具备解构策略的算法通常都涉及到多个简单的子模型而不是一个复杂的大模型，如果能选

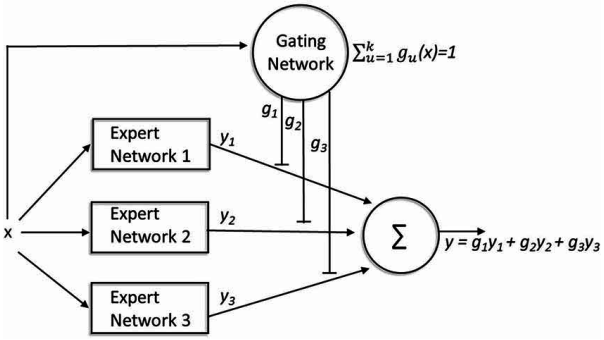


图 2.4　神经网络中的解构策略举例：Mixture-of-Experts 框架[63]。

择适合的子模型复杂度，就能够在方差和偏差之间取得最优的平衡点。

- 对大数据的高扩展性。处理大数据的有效方式包括采样、并行处理、降维、以及解构。解构的目标是将数据的规模分解为普通算法可以处理的规模[64]。
- 提升可解释性。从另一个角度讲，解构操作也可以被认为是将原始的复杂问题分解简化，用简单的模型解决一个个子问题。这使得模型的设计者对问题本身有更直观和清晰的理解[65]。同时，简单的模型也利于数据可视化，进一步方便研究者理解数据、理解问题。
- 实现模块化。模块化使得模型的维护性得到极大的提升。由于数据通常需要经过多轮搜集，模型本身也随着数据的变化、目标的变化而不断迭代。此时，如果模型由多个子模块组成，同时耦合度比较低，那么模型的迭代可以通过重构相关的子模块来实现，维护的难度大大降低[59]。
- 适于并行计算。类似于上一点，当子模块间的耦合度较低，那么每个子模块可以实现并行计算，极大地缩短运算时间。
- 提升模型选择的灵活度。每个子模块（子任务）可以选择不同的模型。例如每个子模块的神经网络层数、初始化方式不同。研究者可以不断探索最佳的模型组合。

2.2.1　路径解构

对于大部分的图挖掘任务来说，节点之间的关系亲疏都是研究的重点。邻接矩阵能够以 one-hot 向量的方式刻画图中节点的一度邻居。显然，这种向量是高维的、稀疏的、离散的。这样的表示向量不适于机器学习模型的计算。在自然语言处理领域，此向量同样存在着相似的问题。2013 年诞生的 word2vec 技术极大地提升了词向量的表示能力。它将原本高维的、稀疏的、离散的词向量转换为低维的、稠密的、连续的表示向量。word2vec 的核心思想是希望词向量可以重构出词与词之间共同出现的概率。有一系列图表示学习算法借鉴了这一思路。而类比的关键就是如何定义图之中节点共同

出现的概率。由此，随机游走的概念自然地被提及。

随机游走模型可以基于图结构产生大量随机路径。通过将节点类比为单词，游走路径类比为句子，文本中词语的共现概率也可以被迁移到图中。具体来说，基于随机游走的图表示学习的目标是“如果两个节点的向量相似，那么它们共同出现在随机游走的同一窗口里的概率也大”。值得注意的是，在这种方法中，两个点的临近度并不是一个确定的值（比如像上一章节中是通过邻接矩阵计算得到的），而是在随机过程中统计得到的一个近似值。这使得模型的鲁棒性更强。

DeepWalk[10] 和 node2vec[66] 是最经典的基于随机游走的图表示学习方法。它的基本思路是希望学到的向量表示和随机游走有这样的关系：

$$p_{\mathcal{G},\mathcal{T}}(v_j|v_i) = \frac{e^{z_i^\top z_j}}{\sum_{v_k \in \mathcal{V}} e^{z_i^\top z_k}} \tag{2.2}$$

其中 p 指的是从 i 点出发的随机游走，在 $\mathcal{T}$ 步之内访问到 j 点的概率，通常，在 $\mathcal{T}$ 的取值范围是 2 到 10 的整数。和上文的临近关系度不同的是，此处的 p 值既是随机的又是不对称的。算法的优化目标是希望最小化的是一个交叉熵：

$$\mathcal{L} = \sum_{(v_i,v_j)\in\mathcal{D}} -\log(p_{\mathcal{G},\mathcal{T}}(v_j|v_i)). \tag{2.3}$$

其中 $\mathcal{D}$ 是训练集，是从图中的每个点出发进行多次随机游走采样得到的。然而直接优化这个目标的计算复杂度是很高的。尤其是解码器的分母部分，需要计算所有节点。因此 DeepWalk 和 node2vec 使用了一些优化的技巧以应对这部分的计算量。优化的技巧包括 hierarchical softmax 和 negative sampling。

DeepWalk 和 node2vec 的不同之处在于它们对于随机游走的控制。DeepWalk 利用的是没有任何限制的随机游走，而 node2vec 引入了两个参数 p 和 q 来控制随机游走的方向。参数 p 控制了随机游走往回走的概率，q 控制了随机游走往远处走的概率。引入了这两个参数之后，随机游走可以在深度优先搜索和广度优先搜索之间平滑的转换。node2vec 的作者声称调节这两个参数可以使得学习出的向量对社区结构或者节点的局部结构特征更加明显。

HARP 通过图预处理的方法扩展基于随机游走的算法。最近 Chen 等人[67] 引入了一个叫做 HARP 的策略，它可以提升很多基于随机游走的算法效果。它首先将图中相似的节点合并成一个超点，图的规模因此缩小很多。在缩小后的图上跑 DeepWalk，node2vec 或者 LINE，为每个节点（普通节点和超点）学习到了一个对应的向量。然后再在原始图上重新跑一遍原算法，以上一步学习得到的向量为训练起始状态。

随机游走算法的其他变种。还有很多算法都考虑对随机游走部分加以控制。比如 Perozzi 等人[68] 进一步扩展了 DeepWalk 算法，在随机游走的过程中有选择性地跳过一

些节点，这样能够得到和 GraRep 算法比较类似的结果。Chamberlan 等人[69] 将 node2vec 的解码器部分修改，将欧式空间中的向量内积更改为双曲空间中的向量内积。

DNGR[70]使用了一个在深度学习中被广泛使用的模型——自编码方法来压缩得到每个节点的局部邻接关系。DNGR 是由多层神经网络堆叠实现的：在编码器的每层中，邻接信息会被一层层的降维；在解码器中，邻接信息会被一层层的升维。DNGR 的解码器目标是恢复基于随机游走的节点邻接关系。

2.2.2 邻域解构

由于现实生活中大量图的形成原因，从本质上讲，与社交有关，而社交存在着同质性、传递性，在研究图中某个节点时，取其所在的邻域能够得到更加丰富和优质的数据。最典型的研究案例是自我中心网络（ego-network）。这种研究关注的是目标节点及其一度邻居组成的导出子图，有时范围也会扩展到二度邻居[71,72]。经典的结构洞理论即与自我中心网络密切相关。Burt 提出了网络的约束效应及其对应的约束系数。该系数需要计算每个节点以自我为中心时与他人相连所受到的约束程度[73]。

在图表示学习中，很多算法都设计了依赖于节点局部连接的表示模型。这些算法背后的思想就是利用节点周围的节点信息和邻接关系去产生节点的向量表示。与之前讨论的问题不同的是，这些邻居节点融合的算法利用的是节点的特征和属性来产生表示向量。邻居节点融合算法利用的就是这类信息。如果出现没有节点属性可以利用的情况，这些算法可以将某些结构特征当作节点属性（如节点度数）[57]，或者给每个节点对应一个 one-hot 编码并将其视作属性[74,75]。这类算法也被称作卷积式的算法，因为它们用一个节点的邻域去代表这个节点，这与计算机视觉中的卷积核有着相同的出发点[76]。

在卷积式的表示学习模型中，邻居节点融合算法以一种循环迭代的方式构建节点的向量表示。节点的向量表示首先用节点的标签属性作为初始化。再接下来的每轮的编码循环中，节点通过一个特殊设计的集合函数将它们邻居的向量表示集合在一起。集合的操作使得每个节点都有了一个新的向量表示，它结合了节点上一轮中的向量和它的邻居的向量。这个新的向量表示会经过神经网络的处理最终得到本轮迭代的节点向量最终表示结果。随着迭代轮数的增长，节点的向量表示会包含在图中越来越远的节点的信息。然而节点的表示维数并没有增长，编码器实际上强制性地把多跳的邻居信息压缩到一个低维空间的向量中。在 K 轮过后，迭代停止，节点在第 K 轮的向量表示就是它最终的节点表示。

近年来有很多算法都是在这种框架之上设计的，其伪代码如图2.5所示。例如图卷积模型（GCN）[56,75,76]，column network[77]，GraphSAGE 算法[57]。在图2.5中涉及到一些学习的参量：集合函数中的参数和权值矩阵 W，它们控制了算法究竟是如何将邻

Input : Graph $\mathcal{G}(\mathcal{V},\mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth K; weight matrices $\{\mathbf{W}^k, \forall k \in [1,K]\}$; non-linearity σ; differentiable aggregator functions $\{\text{AGGREGATE}_k, \forall k \in [1,K]\}$; neighborhood function $\mathcal{N}: v \rightarrow 2^{\mathcal{V}}$

Output: Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$

1 $\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$;
2 **for** $k = 1...K$ **do**
3 　**for** $v \in \mathcal{V}$ **do**
4 　　$\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$;
5 　　$\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{COMBINE}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)\right)$
6 　**end**
7 　$\mathbf{h}_v^k \leftarrow \text{NORMALIZE}(\mathbf{h}_v^k), \forall v \in \mathcal{V}$
8 **end**
9 $\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$

图 2.5　邻居聚合计算框架[1]。

接的节点信息融合在一起的。与直接编码不同的是，这些参数是被所有节点所共有的，即所有的节点共用这一套参数，计算过程中只是每个节点的属性和邻居节点不同而已。参数共用提升了算法的效率（参数的维数与图的大小无关），引入了正则项，同时使得算法可以为那些在模型训练过程中未出现的节点计算出其相应的向量表示。GraphSAGE，column networks 和各种 GCN 方法的大体框架都是图2.5，只是它们有各自不同的 aggregate 和 combine 函数。GraphSAGE 在算法第 5 行的地方用了向量拼接的方法，作者还尝试了按位取平均、max-pooling 和 LSTM 等方式构建 aggregate 函数。他们发现越复杂的 aggregate 函数，越有可能得到好结果。GCN 和 column network 在第 5 行用了加权求和的方法，在第 4 行用了按位取平均的方法。column network 在算法的第 7 行还增加了插值项，这使得模型随着迭代次数的增长，仍能保留一定的局部标签特征。

2.3　本章小结

本章以两条主线分别梳理了经典文献和近年的相关研究。两条主线中所涉及的文献有所交叉。第一条主线关注图的表示方法，依此介绍了图的离散表示：社区发现，以及两种图的连续表示：图表示学习和图神经网络。第二条主线则关注图挖掘算法中的解构策略，分别介绍图挖掘算法中的路径解构方式和邻域解构方式。

第三章　图挖掘任务中的解构策略

章节1.2简述了解构策略的定义和关键研究问题。而本章则试图以较为宏观的角度，剖析解构策略的理论基础、适用范围等相关问题。

3.1　图的解构

3.1.1　节点属性、导出子图与子图解构

以社会心理学的观点来看，“物以类聚，人以群分”展现的是影响人际交往的重要因素：相似性。人们倾向于喜欢那些具有相似的人，并且随着接触机会的增加，喜欢的程度不断加深；反之，不相似的人即使彼此接触，其态度也很难改变[78]。相似性体现在年龄、社会地位、空间距离、价值观、信念等等多个方面。在交往的不同类型和阶段中，人们对这些方面的重视程度各不相同。但总的来说，当讨论参与者特征和图结构的关系时，可以认为“特征相似使得参与者产生连结”。

同样以社会心理学的观点来看，“近朱者赤，近墨者黑”则可以视为同化效应[79]的表现。

定义 4 (同化效应) 当个体面对社会比较信息时，其自我评价水平朝向比较目标的现象。

通俗地讲，同化效应指的是人们的态度和行为逐渐接近参照群体的态度和行为。日常生活中，人云亦云的从众表现、互联网上网民观点的极化现象、集体生活中的情绪传染等等都是同化效应的具体表现。这从另一个角度讨论了参与者特征与图结构的关系：“连结使得参与者的特征趋于相似”。

相似性与同化效应为子图解构的合理性提供了理论支持，解释了节点属性与其图中相邻节点的关系。子图解构针对的是对单一节点属性进行建模、预测的任务。根据章节1.2中子图解构的定义，当需要推测目标节点的属性时，可以不关注整个图，而只基于目标节点的邻域所构成的导出子图进行推测。导出子图的结构、所包含的节点属性都是可资利用的信息，基于此可以分析目标节点与邻域的关系紧密程度，并推测其特征相似程度。具体来说，在图表示方法中，分析每个节点所处的邻域，从中抽取有效信息进行编码，利用邻域的信息生成目标节点的向量表示。在与节点属性相关的具体应用中，利用导出子图进行计算。

3.1.2 节点对关系、随机游走与路径解构

图挖掘任务中，除了单一节点属性的预测以外，节点对之间的关系建模也是一个常见的任务。通常，这类任务通过计算一些简单的节点距离指标（如最短路径距离）来完成，亦或是统计节点之间的路径数目（如计算两个节点的公共邻居数目）来完成。然而，一些工作也注意到这两种方式的缺点。

在真实的图中，信息（或者其他传播内容）并不是沿着预先定义好的路径传播[80–82]。在传播的过程中，新闻或是谣言本身并不知道在图中两点之间的最优路径是怎样的；它们的传播过程更像是漫无目的地随机游走。在著名的小世界实验[83]以及其在不同时代的多个重复实验[84,85]中，参与的人们都被明确要求将信息尽可能快（直接）地传递至目标。然而实验中，这一指令的完成度并不高。这些实例证明，通过最优路径来衡量两点之间的关系是有违实际的。

统计节点之间的路径数目的不足在于其忽视了相关节点自身的重要性（节点度数或类似的中心度）[86]。例如在社交网络中，两对用户都有三个共同关注者：第一对用户的共同关注者是名人，拥有数以万计的关注者；第二对用户的共同关注者是普通人，也许同为第二对用户的同事。按照常理，即便这两对用户的共同关注者数目相等，第二对用户的连结比第一对用户的连结更为紧密。类似的例子不胜枚举。

基于上面的分析，通过短的随机游走路径来衡量节点对关系的优势显现出来。随机游走可以同时体现最优路径长度和路径数目两方面的拓扑信息[86,87]。在图表示学习算法和具体应用中，路径解构将节点对所共享的子图分解为多条随机游走路径，通过分析节点对在随机游走路径窗口内共同出现的概率来刻画节点对的关系。窗口的大小限制了节点对之间路径的最大长度，只有最短和接近最短的路径会被考虑。随机游走的转移概率与每个中转节点的度数相关，也就是说节点对之间的可行路径数目和中间节点的重要性都会影响节点对共同出现的概率。

3.2 解构策略的应用

3.2.1 图挖掘任务中的三大挑战

在图挖掘任务中，关系型数据以图的形式呈现。这类数据有一些原生的特点、难点，使得图挖掘算法在处理它们时不得不面临一些挑战。本章对于图挖掘领域中的重要共性问题进行提炼与总结。

- **同质性形式多样。**同质性指的是“图中，相似的个体（节点）相互连结的概率”。已经有大量的研究证明同质性广泛的存在于社会网络，即便具体的表现形式不完全一致[88]。例如，相似性可能体现在种族[89]，年龄[90]，地理位置[91]，教育水

平[90]，性别[90]，价值观[92]，行为表现等[93]。也就是说，对人们的刻画，从本质上看，是多个维度的，具有多种属性和分类方式。由于同质性潜在地影响着图结构的变化，在图挖掘任务中，预测节点属性标签、连边概率时通常需要基于节点之间相似性的计算。而同质性可能在多个维度上存在，对于图挖掘算法而言，厘清图中的连边和任务所关注的某几个维度之间的关系变得重要而困难。

- **图结构信息缺失**。由于数据采集方式的不足，图数据通常都是不完整的。边、点、以及各类属性信息都有可能缺失[94,95]。例如在时序图中，由于采集频率的限制，一些边和点未能捕捉到即消失；社交网络中，用户信息由于隐私性的规定不是完全可见的。同时有很多结构信息不可被显式地捕捉，但是可以隐式地推测。例如蛋白质关系图中（蛋白质共同参与某个生化反应），某些蛋白质之间的关系还未被生化学家发现；与职场相关的社交网络（如 LinkdedIn）基本上不包含基于家庭的连边。图结构的缺失要求图挖掘算法有更强的鲁棒性和泛化能力。
- **临近度计算复杂**。最短路径、最大流 — 最小割问题、节点中心度度量都是传统图挖掘中经典的问题，其本质是在不同维度上刻画节点之间的临近关系。然而当图的规模不断增大，这些问题的算法复杂度成为了其主要的限制因素。进一步说，这些问题都涉及到节点间路径的计数。大规模图要求更高效、快捷的图挖掘算法。

3.2.2　解构策略有效应对挑战

图1.3中展示了两种解构方式与三大挑战之间的应对关系，接下来的内容将对此展开详细论述。

- **同质性形式多样与邻域解构**

  本文认为，邻域解构可以在一定程度上解决同质性形式多样的问题。在此给出两种具体的应对方式：

  1. 在建模某一节点的特征时，邻域解构截取出节点在图中邻域节点，算法围绕节点与其邻居的关系进行计算。邻域解构从客观上切断了目标节点与图中其他节点的关联，而专注于目标节点的邻域。建模邻域关系的模块可以自由设计，依照不同类型的数据和应用场景，综合利用邻域中包括的多种信息（例如结构、标签、属性等等）。该模块在整个图中共享（即图中每个节点与其邻域之间的关系建模模型是相同的），这使得算法得以挖掘出该图所展现的同质性。
  2. 在一些场景中，同质性体现在节点在图中的地位上，例如同为重要的核心节点或者边缘节点。基于邻域解构截取出的子图可以自然地刻画节点

在图中的地位。需要承认的是，邻域解构也会使得节点的全局结构特征丢失，如中介数一类的指标只能在完整的图中完成计算。在具体的应用场景中需要判断全局特征是否需要。

- **图结构信息缺失与邻域解构**

  邻域解构可以在一定程度上缓解图结构信息缺失的问题。同样，也可以对此给出两种解释：

  1. 邻域解构试图建模目标节点与整个邻域的关系，而不是目标节点与某个邻居节点的关系。少数边的缺失对于邻域的影响不大。通过一定的邻域采样策略（如扩大邻域的划定范围）可以有效地应对边的缺失。即便边的缺失导致邻域的截取错误，目标节点也可以从已截取出的邻域节点中获取相对充足的信息。
  2. 邻域解构为目标节点划定邻域范围后，关系建模模型可以对邻域内的节点一视同仁，而不考虑邻域内具体的连接情况。也就是说，截取的子图被补全为完全子图。数据采集过程中一些未被捕捉到的边可以借由这个机会被补全。

- **图结构信息缺失与路径解构**

  路径解构可以在一定程度上缓解图结构信息缺失的问题。如3.1.2节所述，路径解构将节点对所共享的子图分解为多条随机游走路径，通过分析节点对在随机游走路径窗口内共同出现的概率来刻画节点对的关系。通过控制窗口的大小，可以直接控制两点之间可行路径的范围，不仅只有最短的路径会被考虑，那些比较短的路径也会被考虑。少量边的缺失会导致一些最短路径的改变，但是设置窗口可以为最短路径的波动提供一定空间，增强了算法的鲁棒性。

- **临近度计算复杂与路径解构**

  路径解构主要涉及随机游走的路径的生成和分析。有大量的研究支持随机游走的快速计算。例如，线性优化[96,97]，蒙特卡罗近似[98,99]，矩阵计算[100]等等。除了快速计算，路径解构的优势还体现在不受图规模、图演化的影响。随机游走的生成只与访问节点的度数相关，所涉及的计算与图的整体规模无关。当图演化，结构（包括节点和边）发生变动，最优路径需要针对新图重新计算，随机游走则能够以一种增量式的方式完成计算：只针对那些结构产生重大变化的节点生成新的随机游走路径。

3.3　本章小结

本章重点讨论了解构策略的理论基础，并且详细介绍了解构策略在应对图挖掘算法的挑战时发挥的作用。

第四章　图挖掘中邻域解构策略的运用

本章介绍邻域解构策略在图挖掘任务中的体现。《战国策》中的“物以类聚，人以群分”描述的是网络科学研究中一个广泛存在的现象。一方面，人们会因为相似的背景、喜好聚集在一起，产生连结；另一方面，处在网络之中的人群会相互影响，出现群体极化的现象（即某个问题经过群体决策后，不论在此之前个体的意见是否一致，经过讨论后往往会得到一致结果）。也就是说，图结构的形成与参与者的同质性互为因果，两者之间关系紧密。图挖掘任务中，当需要推测目标节点的属性时，可以不关注整个图，而只通过目标节点的邻域所提供的有效信息来进行推测。本文将这种操作称作邻域解构。邻域解构不仅可以过滤掉繁杂的噪音信息，保留有价值的信息，同时也可以弱化预测结果对图结构的依赖，少量边的缺失不影响最终结果。

本章的第一部分关于图的表示方法，关注节点的离散表示，提出一个运用邻域解构策略的重叠社区发现算法。重叠社区发现算法的目标是将图划分为多个内部联系紧密的节点块（即社区），节点可以同时属于多个社区。重叠社区发现算法的结果可以用来预测节点属性、预测信息传播过程、预测图演化等。本文提出的算法 Fox，从模型功能上讲，将社区发现任务分解为判断一对邻接节点是否同属于一个社区的子问题；从模型所处理的数据上讲，将原始图解构为目标节点的自我网络（*ego-network*）。Fox 算法通过“数三角形”的方式，使得每个节点从邻域中分辨出真正相关的邻居节点，解决了图中同质性形式多样的问题。Fox 算法中，由于多次邻域解构叠加在一起，节点有可能与非相邻的节点同属于一个社区，可以缓解图结构信息缺失的问题。

本章的第二部分关于图挖掘对具体应用，提出利用社区划分的结果来辅助影响力最大化任务。传统模型在计算节点集合在图上的信息传播影响力时，往往很难度量节点之间传播范围的差异性。本文提出的差异性影响力最大化模型 DIM 基于运用邻域解构策略的重叠社区发现算法，利用社区划分结果快速计算节点影响力范围，实现增大传播范围差异性的目标。

4.1　大规模重叠社区发现算法 Fox

4.1.1　研究背景

社区发现是一个基础并重要的研究工作。社区指的是图中相互间连接比较紧密的节点簇。物以类聚，人以群分。对于同一簇的节点来说，节点之间势必在某些方面上相似的，尽管这些相似点对于研究者可能是未知的。通过社区发现工作，研究者们可

以对节点之间的关系有进一步认识，从而对节点的特性进行某些合理推测。重叠社区允许节点同时属于多个社区。在社交网络中，这一概念十分普遍。人们很自然地同时处于多个社交圈子，比如家庭，工作，校友等等。在文献[21] 中，作者指出重叠是真实图中普遍存在的现象。

社区发现在很多领域中引起了研究者的关注。其中大部分的研究关注于离散的社区，并且离散社区发现算法已经在大规模图上有了应用的案例[101–104]。但对于重叠社区发现来说，尽管在现实生活中被应用得更加广泛，其在大规模图上的可扩展性往往不尽如人意。大部分的重叠社区发现算法的目标是发现一些预先设定好的目标图结构或者优化某个与结构相关的计算指标。然而，这两种方式都不能反应社区形成的过程。社区发现的结果好坏与预定义的发现目标或者计算指标的质量直接相关。除非算法设计者们有考虑图及社区的演化特征，大部分的算法很难以一种增量式的方式进行计算。

Clique Percolation[105–107] 将邻接的完全子图视作社区。由该定义可知，这类算法只适用于连接稠密的图。生成式模型包括混合随机块模型（MMSB）[34,108] 和非负矩阵分解（NMF）[35] 法。这类算法的瓶颈在于矩阵乘法的时间和空间复杂度。种子扩展类的算法在重叠社区发现中也占据一席之地[25,33,109–111]，这类算法首先选定一些种子节点，再依据一个扩展策略方程（模块度，导纳等等）来判断节点是否被纳入社区之中。这类算法适用于大规模图，但是往往划分的质量不高。也有利用图表示学习技术来挖掘社区[112,113]。但是，由于社区的探测是发生在学习到的隐空间中，社区的划分结果往往不直观，并且可解释性不强。还有一些重叠社区算法是基于标签传递[114] ，边聚类[115]，组合优化[116]，和非凸优化[117] 等思想。

社区的产生是人类社交活动的产物。人们通常喜欢与相似的人聚集在一起。每个人都自主地选择加入不同群体。整个过程与博弈论的联系不言而喻。在文献[118]中，博弈论的相关理论与社区发现任务首次相结合。与一般的以优化某一具体指标为目标的社区发现算法不同[119]，基于博弈论的算法思路更自然，在不同类型的图中都会有比较稳定的表现。基于博弈论的算法通常受限于算法的可扩展性[120–124]。

4.1.2 模型描述

4.1.2.1 算法简述

本章从一个新的角度 — 博弈论，重新审视社区发现的问题，并提出了一个对大规模数据的快速社区发现算法 Fox （Fast Overlapping Community Search）。每个节点有权利自主地选择加入最合适的社区，节点的选择又相互影响。比如说，节点 a 和 b 选择加入社区 C，他们的好朋友节点 c 因此也想加入社区 C；节点 a，b，c 先后都加入了社区 C，C 中原来的成员 d 和 e 认为它跟 a，b，c 都不熟悉，他们不适合再待在这个

社区中了，因此节点 d 和 e 选择离开社区 C，加入更适合它的社区 D；节点 f 既跟 a，b，c 三个节点很熟悉，又跟 d 和 e 很熟悉，因此 f 选择同时加入社区 C 和社区 D。这个例子说明了节点之间的博弈过程。由于每个节点都不断地做出自己认为最正确的判断，最终整个系统会出现纳什均衡。在均衡状态下每个节点都对自己社区归属状态感到满意，因此不会再做出离开社区的决定。Fox 算法即是基于这样的框架建立起的算法，同时在实际计算中也做出了一定程度上的化简和省略处理，加速整个的计算过程，以适应数据的超大规模。

博弈过程中，Fox 选用了 WCC 指数[125]作为每个节点的效用得失函数 (utility function)。WCC 具有三个突出的特点。一是 WCC 指数的根本原理是计算社区内三角形的个数。在研究[126]中，对于重叠程度很高的图来说，计算三角形个数是最有效的社区发现方式。二是对 WCC 的计算又作了进一步的近似，使得它的计算复杂度降低到 O(1)，从而保证了算法的执行速度。最后一点是 WCC 可以表示单个节点与社区的从属关系对整个社区划分的影响，这符合了博弈论中潜在博弈的基本要求。同时也将三角形的概念扩展为“半三角形”，即三个节点两条边的结构，并基于半三角形提出了 s-Fox 算法。这个算法可以有效扩大搜索出的社区的规模。总的来说，Fox 算法有两个主要贡献：

1. Fox 算法是已知的最快的重叠社区发现算法。尤其对于大规模的图数据，Fox 算法相较于其他任何现存算法有着极大的速度优势。
2. Fox 算法的可扩展性极好。算法复杂度与图的边数几乎呈线性关系。

Fox 算法的模型是基于博弈论中的潜在博弈的。节点（参与者）根据当前的社区划分情况选择加入对自己而言最佳的社区（决策）。最佳的社区就是能给节点带来最大收益（效用方程）的社区。所有的节点都依次作出最佳的决策，最终整个系统达到一个稳定的状态。算法伪代码在 Algorithm1展示。

4.1.2.2　效用函数与潜函数：WCC

首先来介绍一下效用函数 WCC。给定图 $G(V,E)$，V 是节点集合，E 是边集合。对于节点 x 和社区 C 来说，

$$\mathrm{WCC}(x,C)=\begin{cases}\frac{t(x,C)}{t(x,V)}\cdot\frac{vt(x,V)}{|C-\{x\}|+vt(x,V-C)}, & \text{if } t(x,V)>0;\\ 0, & \text{if } t(x,V)=0.\end{cases} \tag{4.1}$$

其中 $t(x,C)$ 表示在社区 C 中 x 和其他节点形成的三角形个数；$vt(x,C)$ 表示 C 中与 x 和 V 中某一点组成至少一个三角形的节点个数；$C-x$ 表示 C 去掉节点 x 后的其他节点组成的集合。$\mathrm{WCC}(x,C)$ 衡量的就是节点 x 和社区 C 的关系紧密程度。简单的说，式子的第一部分描述的是 x 节点所组成的三角形。这一比值越大越好。第二部分描述的

Algorithm 1 Strategy(x)

```
remove ← 0
transfer ← 0
copy ← 0
max ← 0
C ← current community
// Judge whether to stay in the current community.
if Δ_L(x, C) > 0 then
    remove ← 1
end if
// Judge whether to move to the adjacent communities.
for all C in adjacent communities do
    // Judge whether transfer to an adjacent community or be alone.
    if remove then
        if Δ_T(x, C) > max then
            max ← Δ_T(x, C)
            best community ← C
            transfer ← 1
        end if
    else
        // Judge whether copy to an adjacent community or do nothing.
        if Δ_C(x, C) > max then
            max ← Δ_C(x, C)
            best community ← C
            copy ← 1
        end if
    end if
end for
if transfer then
    remove ← 0
end if
```

是与 x 共同组成三角形的其他节点。这些节点应该尽可能多的包含在 C 中。

本章将这个指标扩展到重叠社区检测的任务中，使得一个节点可以从属于多个社区。对于一个节点 x 和一个社区划分 $P = \{C_1, C_2, \ldots, C_k\}$,

$$u_x(P) = \mathrm{WCC}(x) = \sum_{x \in C_i} \mathrm{WCC}(x, C_i) \tag{4.2}$$

$$\Phi(P) = \mathrm{WCC}(P) = \sum_{x=1}^{n} \mathrm{WCC}(x) = \sum_{x=1}^{n} u_x(P) \tag{4.3}$$

效用方程 Δ 被定义为 $\Phi(P) - \Phi(P')$.

本章进一步定义了半三角形。它由三个点和两个边组成。根据三元闭包原理，两个人在拥有共同朋友的情况下很有可能也成为朋友。而半三角形刻画的就是两个人拥有共同朋友的情形。因此半三角形和三角形一样，也可以作为社区评判的一个指标。区

别在于：半三角形可以找到潜在的连接关系并因此扩大社区的规模。在接下来的章节中，统计半三角形的 Fox 算法记为 s-Fox 算法。

4.1.2.3　策略

在每轮迭代中，每个节点拥有 4 个可选的策略：（1）不改变社区，（2）离开现在的社区且不加入任何社区，（3）移动至另一个社区，（4）在现在的社区的同时加入另一个社区。第 4 个策略使社区之间有可能是重叠的。节点可以被视为被复制到了另一个社区之中。每个策略的回报都是由效用方程来衡量的，每个节点都要做出使效用方程取得最大增长 Δ 的决策。下面将分别展示四种策略所对应的 Δ。其中 P 是 x 做决策前的社区划分，P 是 x 作出最优决策后的社区划分。

1. 策略 1 不改变社区：这个社区划分都没有改变，$P' = P$

$$\Delta_S = \mathrm{WCC}(P) - \mathrm{WCC}(P) = 0$$

2. 策略 2 离开社区并不加入任何社区：假设原来的社区划分是 $P = \{C_1, C_2, \ldots, C_k\}$，$x$ 离开社区 C_k 后，$P' = \{C_1, C_2, \ldots, C'_k, \{x\}\}$，其中 $C_k = C'_k \bigcup \{x\}$.

$$\Delta_L(x, C_k) = \mathrm{WCC}(P') - \mathrm{WCC}(P) \tag{4.4}$$

3. 策略 3 移动至另一个社区假设节点 x 从 C_1 移动至 C_k，原始的社区划分为 $P = \{C_1, C_2, \ldots, C_k\}$，新的社区划分为 $P = \{C'_1, C_2, \ldots, C'_k\}$，其中 $C_1 = C'_1 \bigcup \{x\}$，$C'_k = C_k \bigcup \{x\}$. 这个策略实际上是两个步骤的叠加。首先节点 x 离开 C_1 并且未加入任何社区，此时社区划分为 $P_m = \{C'_1, C_2, \ldots, C_k, \{x\}\}$。其次，节点 x 加入 C_k, $P' = \{C'_1, C_2, \ldots, C'_k\}$。可以认为第二步是策略 2 的逆过程。

$$\begin{aligned} \Delta_T &= (\mathrm{WCC}(P') - \mathrm{WCC}(P_m)) + (\mathrm{WCC}(P_m) - \mathrm{WCC}(P)) \\ &= \Delta_L(x, C_1) - \Delta_L(x, C_k) \end{aligned} \tag{4.5}$$

取得最大 $\Delta_L(x, C_k)$ 的社区被定义为最佳移动社区，x 从原社区移动到最佳移动社区的 WCC 增长就是 $\Delta_T(x, C_{best})$。

4. 策略 4 保持在原社区并加入另一个社区假设节点 x 被复制到 C_k 社区，复制前后的社区划分别是 $P = \{C_1, C_2, \ldots, C_k\}$ 和 $P' = \{C_1, C_2, \ldots, C'_k\}$，其中 $C'_k = C_k \bigcup \{x\}$。同样这也是一个复合变换，中间状态是 $P_m = \{C_1, C_2, \ldots, C_k, \{x\}\}$。

类似的，最佳复制社区对应的就是 $\Delta_C(x, C_{best})$。

$$\begin{aligned}\Delta_C &= (\mathrm{WCC}(P') - \mathrm{WCC}(P_m)) + (\mathrm{WCC}(P_m) - \mathrm{WCC}(P)) \\ &= -\Delta_L(x, C'_k) + \mathrm{WCC}(x, \{x\}) \\ &= -\Delta_L(x, C'_k)\end{aligned} \tag{4.6}$$

在以上这 4 个策略中，x 将选择实施那个最大的 WCC 提升的决策。

简单的说，在每轮迭代中，如果 x 是不利于它所在社区的连接紧密性的，它一定将会在本轮离开这个社区。在这种情况下，如果 x 同时也对其他任意一个社区有害，那么 x 将成为一个独立的点，不从属于任何一个社区。如果 x 可以增进某个或某几个社区的连接紧密性，那么 x 选择加入带来益处最大的那个社区。如果 x 是利于自己所在社区的连接紧密性的，那么它可以考虑是否复制加入其他社区。x 一定为复制加入的社区带来正向的影响，且该社区是所有社区之中正向的增量最大的那一个。

4.1.2.4 迭代计算

预处理

预处理是指社区划分的初始化。预处理阶段利用了局部集聚系数 (local clustering coefficient，CC) 来作为初始划分的依据，这与算法的主要迭代过程是十分相匹配的。图中一个节点的局部集聚系数衡量了它和它的邻居节点与一个完全图的差别。可以认为，一个节点的局部集聚系数越大，它和它的邻居节点越有可能是一个社区。

预处理中首先计算了每个节点的 CC 值，并按降序排列。如果两个节点的 CC 值相同，那么度数大的节点排在前面。随后将排在首位的节点及其邻居节点挑出，作为一个社区。再从剩下的节点挑出排在首位的节点及其邻居节点，组成第二个社区。如此处理，直至所有的点都被挑走。显然，这个初始划分是不重叠的社区。

多轮博弈

根据上面的分析，可以发现 $\mathrm{WCC}_L(x, C)$ 在整个的决策过程中扮演者至关重要的作用。WCC_T 和 WCC_C 都是在 WCC_L 的基础之上计算出来的。但当计算每个节点形成的三角形个数时，可以想象所涉及的巨大运算量，尤其对于密集连接的图来说计算量更是大的惊人。如果节点数目是 n，平均度数是 d 的话，计算 WCC_L 的复杂度是 $\mathrm{O}(nd^2)$。本章从统计的观点出发，提出了一种近似计算的方法，将这一步的复杂度显著地降低到 $\mathrm{O}(1)$。

在前面的章节中，$\mathrm{WCC}(x, C)$ 的计算方法已经被展示。当节点 x 在社区 C 外部时，三角形数目近似的计算为

$$\hat{t}(x, C) = \binom{|d_C|}{2} \cdot p \tag{4.7}$$

$$\hat{t}(x, V-C) = \binom{|V-C|}{2} \cdot cc \tag{4.8}$$

$$\hat{vt}(x, V-C) = d_{V-C} \tag{4.9}$$

其中 V 是图中的节点全集，p 是两节点在社区中相连的概率，d_C 是 C 和 x 之间边的数目，图的聚集系数是 cc。当 x 在社区 C 内部时，近似计算是类似的。总的来说，近似计算基于三个假设 (1) 社区内部，节点相互连接的概率是相等的 (2) 在稠密的图中，每个边至少参与了一个三角形 (3) 所有节点的局部聚集系数是相同的。

继续对**策略 2** 的分析，新旧两种社区划分的区别是 x 的离开。只有节点 x 和 C_k 内部节点的 WCC 值发生了变化。因此只需要考虑这部分节点的变化。

$$\begin{aligned}\Delta_L(x, C_k) = &\sum\nolimits_{n \in C'_k} (\mathrm{WCC}(n, C'_k) - \mathrm{WCC}(n, C'_k \bigcup \{x\})) \\ &- \mathrm{WCC}(x, C'_k \bigcup \{x\})\end{aligned} \tag{4.10}$$

对于 C'_k 中的所有节点来说，它们可以被划分为两类。第一类节点 N 是 x 的邻居，而第二类节点 M 不是。接下来，可以分别计算不同种类的节点的 WCC 变化。

$$\begin{aligned}\Delta_L(x, C_k) &= \sum\nolimits_{n \in N} (\mathrm{WCC}(n, C'_k) - \mathrm{WCC}(n, C'_k \bigcup \{x\})) \\ &+ \sum\nolimits_{n \in M} (\mathrm{WCC}(n, C'_k) - \mathrm{WCC}(n, C'_k \bigcup \{x\})) \\ &- \mathrm{WCC}(x, C'_k \bigcup \{x\}) \\ &= |N| \cdot \Delta(a) + |M| \cdot \Delta(b) - \mathrm{WCC}(x, C'_k \bigcup \{x\})\end{aligned} \tag{4.11}$$

其中 $\Delta(a)$ 是 N 类节点的 WCC 变化均值，$\Delta(b)$ 是 M 类节点的 WCC 变化均值。

$$\Delta(n) = \mathrm{WCC}(n, C'_k) - \mathrm{WCC}(n, C'_k \bigcup \{x\})$$

接下来分别讨论三类节点：N 类节点，M 类节点和节点 x。在每轮迭代前，需要计算如下统计量：

- d_{in}: x 和 C_k 之间的边数
- d_{out}: x 和 G-C_k 之间的边数
- p_{in}: C_k 中两个节点相连接的概率
- p_{ext}: 图的集聚系数
- q: C_k 中的点与 G-C'_k-$\{x\}$ 中的点之间的平均边数
- S: C'_k 的节点个数
- p: 全图的平均度数

基于这些统计量，可以近似的计算 $\Delta_L(x, C)$：

$$\Delta(a) = \frac{(d_{in}-1)p_{in}}{0.5(S-1)(S-2)p_{in}^3 + (d_{in}-1)p_{in} + q(S-1)p_{in}p_{ext} + 0.5S(S-1)p_{ext} + d_{out}p_{ext}} \frac{(S-1)p_{in}+1+q}{S+q}$$

$$\Delta(b) = -\frac{0.5(S-1)(S-2)p_{in}^3}{0.5(S-1)(S-2)p_{in}^3+q(q-1)p_{ext}+q(S-1)p_{in}p_{ext}}\frac{(S-1)p_{in}+q}{(S+q)(S-1+q)}$$

$$\mathrm{WCC}(x, C_k' \bigcup \{x\}) = -\frac{(d_{in}(d_{in}-1)p_{in})(d_{in}+d_{out})}{(d_{in}(d_{in}-1)p_{in})+d_{out}(d_{out}-1)p_{ext}}\frac{1}{S+d_{out}}$$

4.1.2.5 迭代终止条件及后处理

节点依次做出决定无可避免地会带来一个问题：社区的连通性。很惊讶类似的问题并没有在相关研究中被提及[122–124,127]。如图4.1所示，节点的颜色代表了他们所属的社区。其中 *a* 图代表在某一迭代轮次中的初始划分状态。假设在算法实现中，节点编号大的节点先做出决定。10 号节点和 9 号节点率先选择加入 5 号节点的社区。然而，轮到 5 号节点的选择时，5 号决定加入 6 号节点的社区。在 *b* 图中，本轮迭代最终的结果被展示出来。可以看到 10 号节点和 9 号节点与社区之中的节点并无连结。也就是说这些被“误导的”节点使得最终划分出的社区不是连通的。

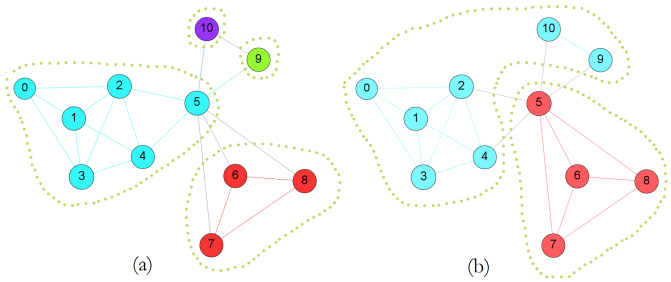


图 4.1 非连通社区划分结果的产生。(a) 和 (b) 是 k^{th} 轮迭代和 $(k+1)^{th}$ 轮迭代社区划分的快照。

潜博弈可以保证算法收敛至纳什均衡点，每个节点都对它当前所处的社区感到满意。在算法迭代的轮次比较多时，大部分的节点都保持稳定，选择不离开当前社区，而小部分被“误导的”节点选择变化社区。接下来的实验中显示：发现这类节点对于最终的社区划分结果实际上影响微乎其微。也就是说，算法不一定要达到精确的纳什均衡点，而在一个近似位置即可。因此 Fox 算法在每轮迭代后都会计算一次 $\widehat{\mathrm{WCC}}(P)$ 值，如果接连两次迭代的 $\widehat{\mathrm{WCC}}(P)$ 变化小于阈值 t，算法终止。

同时，本节提出两个针对被误导节点的处理方式。首先，在每轮迭代中，根据节点度数安排节点决策的次序，度数大的节点先决策。通常度数大的节点有更大的影响力（正如图4.1中 5 号节点所展示的那样）。第二个措施是在算法迭代终止后，对每个社区的连通性进行判断，并对非连通的社区进行标记。这类社区中的节点随后会再次从连通的社区中选择最合适的一个加入。这两个对策可以使得被误导的节点尽可能的少，并且最终的划分结果都是连通的社区。

还有一些小的社区实际上是完全包含在大社区中的。这类社区在最终的社区划分结果中也会被剔除。

4.1.3　快速算法的合理性

为了证明启发方程 $\widehat{\text{WCC}}$ 与 WCC 在最终效果上是十分相似的，本文设计了一些实验加以证明。Fox-naive 算法的 $\Delta_L(x, C_k)$ 是利用 $WCC(P)$ 精确计算的。在图4.2 和图4.3中，可以发现在三个小数据集上，Fox-naive 和 Fox 并没有显著的区别。因此启发方程 $\widehat{\text{WCC}}$ 可以在一定程度上认为是可以依赖的。同时，这两个算法在运算效率上的表现相差巨大。在稍大的 Youtube 数据集上（包含超过一百万节点和两百万边），Fox-naive 花费了 3 天时间。相反，Fox 仅仅用了 8 分钟。

4.1.4　算法复杂度

在预处理阶段，算法复杂度为 $\mathrm{O}(nd^2)$。如果 Fox 算法用于优化一个已经存在的社区划分，这一步可以被忽略。博弈迭代是 Fox 算法和 s-Fox 算法的核心部分，此处的计算复杂度依赖于图中的节点个数和图的连接紧密程度。对于每个节点来说，完成每种策略的 WCC 的增益计算仅仅需要 $\mathrm{O}(1)$。假设图中，节点的平均度数是 d，每个节点需要依次考虑与 d 个邻居节点之间的关系，复杂度为 $\mathrm{O}(d)$。每个节点还需要 $\mathrm{O}(d)$ 来更新自身的社区属性信息，以支持下一轮迭代计算。在每一轮迭代中，所有节点依次做出选择，总的时间复杂度为 $\mathrm{O}(dn) = \mathrm{O}(m)$，其中 n 为节点个数，m 为边的数目。每一轮迭代后，更新全局的统计信息也需要 $\mathrm{O}(m)$ 的复杂度。总共，每一轮的时间复杂度为 $\mathrm{O}(m)$。

4.1.5　带权图上的 Fox 算法

在上一章节中，数三角形个数的操作是在无权无向图上被定义的。然而，在真实的图中，带权图可以更加细致地描述节点之间的连结紧密程度。在这一节中，Fox 算法被扩展到带权图中，完成带权图的社区发现工作。扩展后的新 Fox 算法可以在带权图和无权图之间进行无缝的转换。更确切地讲，对于无权无向图来说，Fox 算法将每一条边视为权重为 1 的双向边。带权图上的 Fox 算法可以得到与无权图上的 Fox 算法相同的划分结果。

首先扩展 WCC 的定义，使之可以应用到带权图上：

$$\mathrm{WCC}_w(x, C) = \begin{cases} \bar{w}_{xC} \cdot \frac{t(x,C)}{t(x,V)} \cdot \frac{vt(x,V)}{|C-\{x\}|+vt(x,V-C)}, & \text{if } t(x,V) > 0; \\ 0, & \text{if } t(x,V) = 0. \end{cases} \tag{4.12}$$

$$\widehat{\mathrm{WCC}}_w(x, C) = \begin{cases} \bar{w}_{xC} \cdot \frac{\mathbb{E}[t(x,C)]}{\mathbb{E}[t(x,V)]} \cdot \frac{deg(x,V)}{|C-\{x\}|+deg(x,V-C)}, & \text{if } t(x,V) > 0; \\ 0, & \text{if } t(x,V) = 0. \end{cases} \tag{4.13}$$

$\bar{w}_{xC}$ 是节点 x 与社区 C 之间的平均边权重。这个定义还可以随着具体的应用的场景变

换。对于带权有向图来说，公式4.11改写如下：

$$\begin{aligned}\Delta_L(x,C_k) &= \bar{w}_{C'_k}\cdot\Big(\sum\nolimits_{n\in N}(\widehat{\mathrm{WCC}}(n,C'_k)-\widehat{\mathrm{WCC}}(n,C'_k\bigcup\{x\}))\\ &\quad+\sum\nolimits_{n\in M}(\widehat{\mathrm{WCC}}(n,C'_k)-\widehat{\mathrm{WCC}}(n,C'_k\bigcup\{x\}))\Big)\\ &\quad-\bar{w}_{xC'_k}\cdot\widehat{\mathrm{WCC}}(x,C'_k\bigcup\{x\})\\ &=\bar{w}_{C'_k}\cdot(|N|\cdot\Delta(a)+|M|\cdot\Delta(b))-\bar{w}_{xC'_k}\cdot\widehat{\mathrm{WCC}}(x,C'_k\bigcup\{x\})\end{aligned} \tag{4.14}$$

其中 $w_{C'_k}$ 是社区 C'_k 的平均边权。对于时间复杂度的分析，带权图上的 Fox 算法与原算法保持一致。

实际上，上述的变化意味着节点之间邻近性的定义得到了扩展，从判断两点之间是否存在边（离散值）变为边权重的大小（连续值）。在这种定义下，节点会在两个情况下被认为与社区连结紧密：节点与该社区可以形成大量三角形，或者节点与社区之间有很多大权重边。从另一个角度说，社区中的每个节点都对是否支持目标节点加入社区有一定的投票权，而边权就代表了该投票权。

4.1.6 实验评测

4.1.6.1 实验准备

实验部分将从两方面展开。一是在有社区标签的小型图上的实验。二是在真实的大型图上的实验。所用的计算机配置是两个 Intel（R）Xeon（R）CPU E5-2620 at 2.00GHz 以及 64GB of RAM。在 Fox 和 s-Fox 算法中，算法结束的阈值 t 取为 0.1%。

真实的大型图

大部分的社区发现算法在实验部分都是用了 SNAP 提供的数据集[128]。本实验从它提供的众多数据集中选择了 3 个具有代表性的小数据集：DBLP 合著图、亚马逊共同购买的商品记录、Youtube 中的兴趣图。在 DBLP 合著图中，节点是论文作者，边代表作者之间的合著关系。倘若一篇论文由 n 名作者共同完成，则 n 个节点组成一个完全连接子图。所著论文入选的会议作为社区的标签。在亚马逊共同购买的商品记录中，图节点是商品，边代表两件商品曾被用户共同购买过。商品的类别作为社区，比如清洁用品、食品、装饰品等等。Youtube 中的用户可以选择自己的关注点，比如科技、娱乐、新闻等等，加入不同的兴趣小组。同时用户之间也可以相互关注。因此在 Youtube 图中，节点是用户，边是关注的关系，社区是兴趣小组。因为选取的数据集规模并不大，因此实验选取的两个对比基线算法可以很好的处理这三个数据集。BigCLAM 算法需要输入检测的社区数目，本实验中将该值就设定为标签集合中设计的社区数目。

可以认为，所谓的社区标签仅仅是一个被大多数人接受的社区划分方式。其实社区探测问题并没有一个标准的答案，毕竟社区本身的定义都还是模糊的。在能找到一

个更好的评价指标之前，这些带有社区标签的数据集可以作为社区检测算法的一个统一评价方法。

本章提出的算法能够有效地解决大规模图中社区发现的问题，并且兼顾速度与正确率。实验总共分析两个大型图数据集：一个移动通信图数据，一个 Google+ 中的相互关注图。移动通信图是中国某市市民的移动通信数据。它包含了超过 400 万用户三个月的通话记录。在图中，节点代表用户，边代表用户之间的通话关系。为了过滤掉垃圾电话的影响，把连边的条件设定为：用户双方都给对方打过 1 次或多于 1 次电话。Google+ 图的规模更大。文献[129]中给出了 Google+ 的一个弱连通分支（weakly connected component），它包含了自 2011 年 7 月到 2011 年 10 月之间 70% 的 Google+ 用户。本章认为用户的相互关注关系从某种程度上来说可以很好的反应他们在现实中熟悉的关系。因此实验中只保留了原始数据中相互关注的关系作为 Google+ 用户之间的边。

考虑到以上的一系列数据操作可能会对图的连通性造成影响，在实验前先计算了两个数据集的最大连通子图的节点覆盖率。结果显示，处理过后的图仍然有很好的连通性。数据如下表所示。N 代表节点数，E 代表边数，D 代表节点平均度数，D_{max} 代表最大节点度数，C 代表最大联通子图包含节点的比例。

表 4.1　数据集的基本信息。

数据集	N	E	D	D_{max}	C	N_c	CC	节点	边	社区
DBLP	317K	1M	6.62	343	100%	13K	0.63	作者	合著	发表刊物
Amazon	335K	926K	5.53	549	100%	75K	0.40	商品	共同购买	商品品类
Youtube	1.1M	3.0M	5.27	28754	100%	8K	0.08	用户	关注	兴趣组
通话图	3.9M	20.5M	10.25	438	94%	-	0.12	用户	通话	-
Google+	22.5M	127.3M	9.98	7347	94%	-	0.24	用户	关注	-

N: 节点数目 E: 边数 D: 平均度数 D_{max}: 最大度数 C: 最大连通子图的节点占比 N_c: 社区数目 CC: 聚集系数 M: 百万 K: 千

对比算法实验中所涉及到的对比算法在表中展示，包括离散的社区发现算法（斜体）和重叠社区发现算法。这些算法所依赖的理论依据展示在表的第三列，算法复杂度展示在第四列。

评价指标

对于所有带有社区标签的数据集来说，实验的评测指标是探测社区与标记社区之间的相似度。两种相似度包括：**F1** 值[35]和**标准互信息**（NMI)[130]。F1 值的计算基于社区成员的准确率和召回率。对于两个社区 A 和 B 来说，

$$H(a,b) = \frac{2ab}{a+b},$$

$$F_1(A,B) = H(\text{precision}(A,B), \text{recall}(A,B)).$$

表 4.2　基线算法的基本信息。

算法	原理	复杂度	引用文献
Louvain	模块度	$O(n^2)$	[25]
Infomap	信息论	$O(n^2 \log n)$	[28]
SCD	启发式算法	$O(mk)$	[127]
SVI	随机块模型	$O(cnk)$	[34]
GAME	博弈论	$O(m^2)$	[123]
BigCLAM	非负矩阵分解	$O(cn+m)$	[35]
OSLOM	局部优化	$O(n^2)$	[33]
nise	种子扩展	与社区的大小相关	[111]
new LFK	局部优化	$O(knd)$	[110]
Fox-naive	数三角形	$O(kmd)$	Sec. 4.1.3
Fox	数三角形	$O(kn)$	-
s-Fox	数三角形	$O(kn)$	-

n: 节点数目 m: 边数目 c: 社区数目 k: 迭代次数

$$\text{precision}(A,B) = \frac{|A \cap B|}{|A|}.$$

$$\text{recall}(A,B) = \frac{|A \cap B|}{|B|}.$$

对于某个社区 p 和一个社区划分 P 来说:

$$F_1(p,P) = \arg\max F_1(p_i,p_j), p_j \in P = \{p_1,p_2,...,p_n\}.$$

两个社区划分 P_1 和 P_2 的 F1 值定义为：

$$\bar{F}_1(P_1,P_2) = \frac{1}{|P_1|}\sum_{p\in P_1} F_1(p,P_2) + \frac{1}{|P_2|}\sum_{p\in P_2} F_1(p,P_1).$$

NMI 值定义为：

$$\text{NMI}(P_1,P_2) = \frac{2I(P_1,P_2)}{H(P_1)+H(P_2)},$$

其中 $H(X)$ 是 X 的熵，$I(P_1,P_2$ 是互信息。F1 值可以从节点角度上评价社区的划分质量。NMI 是从社区角度评价社区划分质量。这两个指标的取值都是 [0,1]，其中 1 代表完全匹配。

对于通话图和 Google+ 图来说，社区标签并不存在。因此可以通过如下几个指标

来评价社区划分质量。**密度**是社区中任意两点相连的概率。

$$\texttt{Density} = \frac{1}{k}\sum_{i=0}^{k}\frac{2m_i}{n_i(n_i-1)}$$

其中，k 是社区的个数，n_i 和 m_i 分别是第 i^{th} 个社区的节点个数和边数。

$\boldsymbol{w_c/w_i}$ 用来评价通话图的社区划分结果[131]。对于通话图来说，并没有一个标准社区划分来辅助判别社区划分的好坏。在文献[131]中，作者提出了一个衡量通话图社区划分好坏的指标。通常可以认为，通话的时长可以用来衡量通话双方的关系亲密程度。人们通常会给自己的工作伙伴和朋友打很多通电话。而在家长和孩子的通话中，通话的时间往往不长，但频率是很高的，因此累计时间也是一个比较大的值。而对于那些与人们生活没什么交集的人，通话时间比较短。在接下来的实验中，定义 w 为边权值，用来表示两个节点所代表的用户之间的通话累积总时长。w_c 代表社区内部的边权值的平均值。w_i 代表跨社区的边的权值平均值。若 w_c/w_i 的值较大，则说明社区内部的通话强度是高于跨社区的通话强度的，社区划分的是较为合理的。实验中可以认为这个评价方式从某程度上讲可以作为通话图的标准划分。

模块度最初在文献[132]中被提出。随后在文献[13]中扩展为重叠社区的模块度。

$$Q_{ov} = \frac{1}{2m}\sum_{c\in C}\sum_{i,j\in V}[r_{ijc}A_{ij} - \frac{s_{ic}k_i s_{jc}k_j}{2m}]$$

4.1.6.2　重叠社区发现算法效果对比

图4.2和图4.3展示了本章提出的算法 (包括 Fox 和 s-Fox), Fox-naive 和其他对比算法的表现。GAME 算法在 Amazon 数据上运行 3 天后，仅仅完成 5% 的社区检测任务（GAME 算法自身支持进度的估算）。因此将其提前终止，并且未将 GAME 纳入下面的结果展示。总的来说，Fox 和 s-Fox), Fox-naive 能力相差并不大。Fox-naive 的效率极差。以 Youtube 数据集为例，Fox-naive 花费了 3 个小时完成计算。而 Fox 仅仅花费了 8 分钟。由此可见，Fox 在没有牺牲效果的条件下，极大地提高了算法的效率。在 F1 和 NMI 两个指标上，Fox-naive 表现都是最好的，但它相对于 Fox 算法的优势基本可以忽略不计。s-Fox 和 OSLOM 算法紧随 Fox 的分数，差距同样并不大。SVI，BigCLAM 和 nise 算法需要设定社区数目。实验中尽可能将其设定为真实的社区数目。然而，对于 SVI 算法来说，真实的社区数目太大，无法完成计算。因此不断尝试更小的可行社区数目，最终将社区数目定为 1000。

人们组成社区的原因来自于两点：相同的身份和相同的连结。对于本实验涉及到的数据集和社区划分来说，相同社区内的节点具有相同的身份，但并不意味着他们之间有紧密的结构。然而，从另一个角度来说，几乎所有的社区划分算法关注的都是图结构，挖掘出与连结结构相关的社区。由此可以解释几个算法在 Youtube 数据集上的

表现都很差。Youtube 的兴趣社区中，成员可能并不是相互关注的，他们仅仅是拥有相同的关注点而已。因此兴趣社区并不能很好的反应出图结构上的紧密连接。本节认为，社区只是一个被人们广泛接受的概念。但鉴于社区本身的定义是不确定的，社区划分问题本身也并不具有标准答案。在找到其他更好的评价指标之前，利用这些带有社区标签的数据集能在一定程度上反应算法效果。

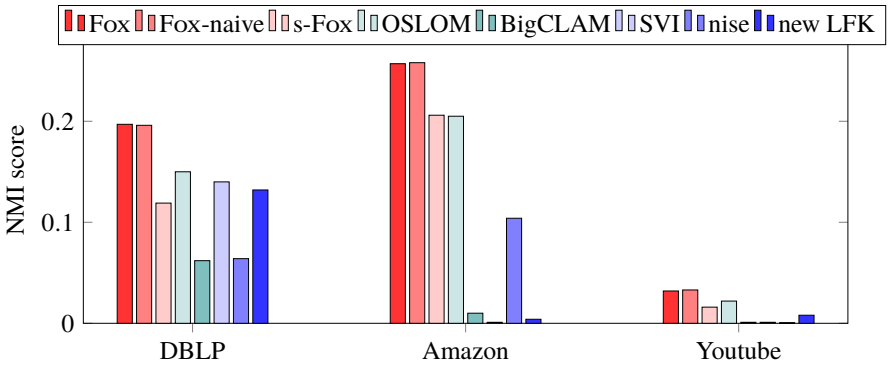


图 4.2　在带社区标签数据集上的 NMI 值。

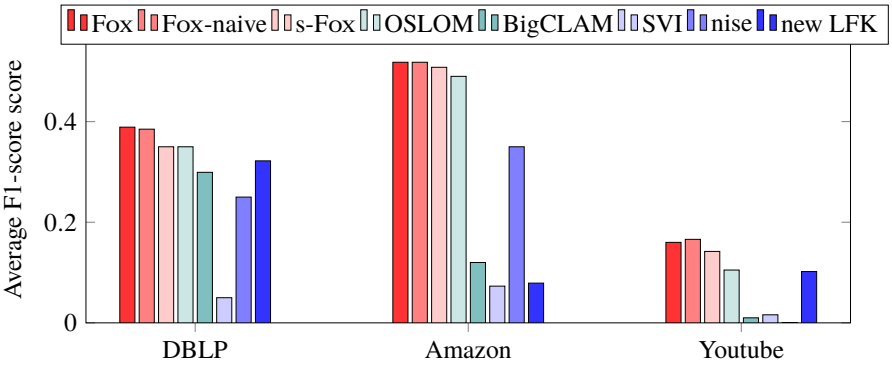


图 4.3　在带社区标签数据集上的 F1 值。

4.1.6.3　改进已有的社区划分

给定某个由其他算法给出的社区划分，Fox 算法可以在其基础之上改进。为了测试 Fox 算法在这个任务上的能力，实验首先使用三个离散社区发现算法对数据集图进行划分，再使用 Fox 算法更新。提升的比例如表4.3所示。SCD 算法在 Youtube 数据集上的表现已经很好，Fox 算法没有进一步提高，因此在表中提升率为 0。Infomap 算法在 Amazon 数据上表现很差，得到的每个社区都不是连通的。因此，Fox 算法也很难在此基础之上提升划分结果。除此以外的其他案例中，Fox 算法都可以取得较大的划分结果提升。

表 4.3　Fox 提升离散图发现算法的结果。

	NMI 值			F1 值		
	DBLP	Amazon	Youtube	DBLP	Amazon	Youtube
Louvain+Fox	38%	22%	100%	9.6%	3.8%	9.9%
SCD+Fox	24%	0.7%	0	4.7%	1.1%	0
Infomap+Fox	18%	-	1.5E+12	77%	-	29%

表 4.4　通话图中社区检测结果的基本信息。

算法	社区数目	平均节点数	节点数中位数	重叠率	覆盖率
SVI	-	-	-	-	-
GAME	-	-	-	-	-
nise	-	-	-	-	-
new LFK	-	-	-	-	-
BigCLAM	150,000	106.106	107	4.535	0.895
OSLOM	183,854	16.138	14	1.203	0.629
Fox	409,894	14.402	12	2.014	0.661
s-Fox	589,452	18.074	17	2.790	0.999

4.1.6.4　在大规模图上的实验

在上表所示的结果统计表中，本章提出的算法可以在 w_c/w_i 一项上取得最好的成绩，同时时间花费又是最少的。尽管在社区发现阶段，算法并没有把边权值考虑在内，但是在最终的检测结果中 w_c/w_i 仍然取得不错的成绩。因此推断，当图中边权值与节点的连接紧密程度成近乎正比的关系时，本章提出的算法同样适用于带权图。边密度（edge density）被定义为 $2m/nn-1$，其中 m 是社区内部的边数，n 是社区的节点数目。社区规模较小时，更容易取得更大的边密度。注意到 OSLOM 与 s-Fox 相比，有着更大的平均大小，同时也有着更大的边密度。经过更加深入的分析，发现 OSLOM 算法给出了大量的三角形社区检测结果，占到了社区总数的 13.1%，而三角形在 s-Fox 社区中的比例仅仅是 2.5%。三角形社区的边密度一定是 1。这为边密度的平均值的提升做出了很大的贡献。因此 s-Fox 有着小一些的社区平均大小和边密度也是可以理解的。还有一点需要说明的是，BigCLAM 算法可以自主决定检测社区的数目，也可以人为的输入社区数目。但是在通话数据上，BigCLAM 自主决定的社区数目较大，以至于它自身都完不成如此庞大的社区检测任务。本实验人为的实验了几个社区数目，而 150000 是算法可以完成社区检测任务的最大社区数目。本实验同时还计算了重叠率（overlap ratio）。重叠率的定义是每个节点平均属于多少个社区。在 BigCLAM 中，每个节点平均属于 4.5 个社区，而某个节点最多属于了 453 个社区。这是非常不合理的。在 OSLOM 中，每个节点平均属于 1.2 个社区，重叠度并不高，重叠社区的重叠属性表现的并不明

表 4.5　通话图社区检测结果评价。

算法	耗时	Density	w_c/w_i	Q_{ov}
BigCLAM	38 hr.	0.028	0.604	**1.401**
OSLOM	194 min	0.358	1.810	0.621
Fox	**41 min**	**0.425**	2.130	0.829
s-Fox	43 min	0.185	**4.296**	1.045

表 4.6　Google+ 社区检测结果基本信息及评价。

算法	社区数目	平均节点数	节点数中位数	重叠率	覆盖率	耗时	Density	Q_{ov}
Fox	2,141,638	15.337	13	2.076	0.587	533 min	0.503	1.263
s-Fox	2,770,401	22.074	19	2.819	0.961	464 min	0.198	1.863

显。本章提出的算法给出了较为合理的重叠度，每个人属于 2 到 3 个社区结构中，这与人们在现实生活中的几大圈子 — 家人圈子、工作圈子、朋友圈子也是相对应的。

表4.5展示了通话图上社区划分的质量。Fox 是最快的社区划分算法。在 w_c/w_i 指标上，Fox 和 s-Fox 都比 BigCLAM 和 OSLOM 算法表现优异。OSLOM 的社区划分结果中包含大量的三角形（一个三角形是一个社区），比例高达 13.1%。然而在 s-Fox 这一比例只有 0.3%。在计算密度时，三角形的密度为 1，因此 OSLOM 在密度上可以取得较高的分数。BigCLAM 在模块度上取得不错的分数，但在其他指标上，分数很低。

Google+ 数据的规模大概是通话图的 7 倍本章提出的算法能在 4 个小时内完成社区检测任务。其他的算法都无法处理如此量级的数据。OSLOM 算法在运行超过 3 天后异常中止了。在表格 1 中，通话数据和 Google+ 有着相似的平均度数。而在表格4.6中，在 Google+ 图上的用时约是通话图的 15 倍。这也再次证实前文所述的计算复杂度以及本章提出的算法的可扩展性。

4.1.6.5　算法的收敛性

图4.4和图4.5展示了 Fox 算法在通话图的每一轮迭代中，选择不同策略的节点个数以及模型的优化目标 — $\widehat{\mathrm{WCC}(P)}$ 的值。从中可以观察到算法的收敛状况。在图4.4中，*remove* 代表策略 2；*transfer* 和 *insert* 代表策略 3（*insert* 是策略 3 中的特殊情况，描述的是一个孤立节点加入社区）；*copy* 是策略 4.

如图4.4展示，选择 *transfer* 的节点个数收敛至 0。选择其他策略的节点收敛至接近 0 的位置。也就是说，大部分的节点都对最终所处的社区感到满意，不会再有改变社区所属的需求。有少量被误导的节点可能会选择 *copy* 操作，并在接下来的迭代在被 *remove*。这一现象已在章节4.1.2.5中详细分析过。

选择 *insert* 和 *remove* 的节点数目首先上升，随后逐步下降。原因是最初的几次迭

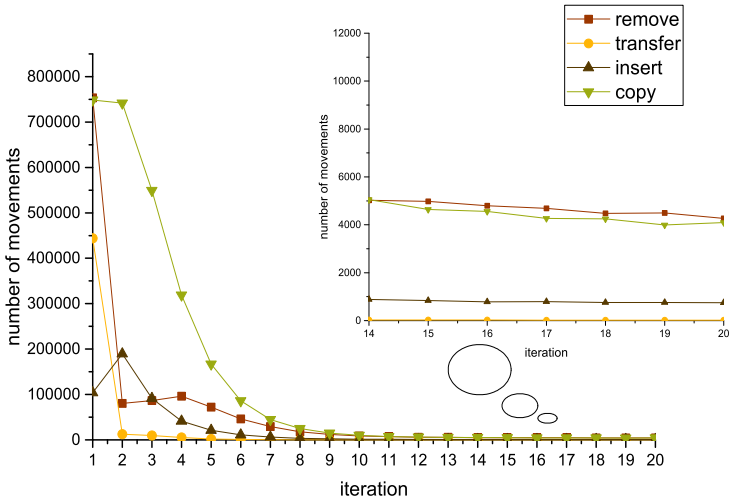


图 4.4　在通话图上应用 Fox 时，随着迭代次数的上升，节点的不同种类移动次数极速下降。

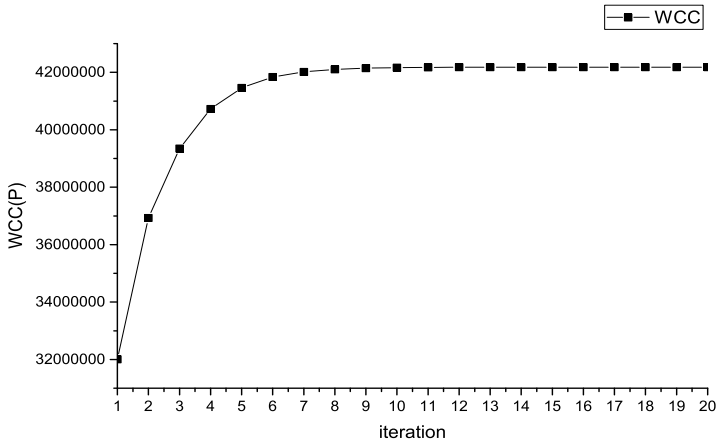


图 4.5　在通话图上应用 Fox 时，随着迭代次数的上升，$\widehat{\mathrm{WCC}(P)}$ 的值迅速达到稳定状态。

表 4.7　不同迭代停止条件下，Fox 在通话图上的表现。未忽略边权

t	社区数目	时间（迭代次数）	社区大小	Density	w_c	w_i	w_c/w_i
0.15	428,310	10min(2)	10.90	0.503	25.32	12.41	2.04
0.10	420,578	15min (3)	12.75	0.455	25.19	12.06	2.09
0.05	415,486	19min (4)	13.65	0.437	25.20	11.93	2.11
0.02	412,560	25min (5)	14.06	0.430	25.22	11.89	2.12
0.01	411,068	28min (6)	14.25	0.427	25.23	11.87	2.12
0.001	409,894	41min (9)	14.40	0.425	25.24	11.86	2.13
0.0001	409,723	58min (13)	14.42	0.425	25.24	11.86	2.13
0.00001	409,723	58min (13)	14.42	0.425	25.24	11.86	2.13

代中，大部分的节点都选择加入其他社区，这一方面造成了社区规模的巨大增长，另一方面也意味着有很多节点是盲目的跟随。在接下来的迭代中，这类节点会渐渐看清状况，离开社区。

图4.5展示的 $\widehat{\mathrm{WCC}(P)}$ 的变化显示，在迭代开始时，$\widehat{\mathrm{WCC}(P)}$ 已有比较高的起点，说明了初始化的有效性。在 7 轮迭代后，$\widehat{\mathrm{WCC}(P)}$ 达到了稳定的状态，说明模型的效率极高。

4.1.6.6　迭代停止条件

实验中设定了不同的迭代停止条件，即阈值 t 的值，并且观察在有权图和无权图上算法的表现。如表4.7所示，随着 t 的减小，迭代的次数在增加，社区的数目也在增加。从社区密度的角度来看，社区越来越大，内部连接紧密程度越来越低。社区内部的边权均值，即 w_c，首先降低，三轮迭代后又逐步上升。与之相反，社区之间的边权均值，即 w_i，在单调地下降，二者的比值 w_c/w_i 在不断地上升。这一趋势表明 Fox 算法可以成功地估计节点与社区之间的关系，并以此为基础不断优化 w_c/w_i 的值。另外，表中也显示当 $t > 0.01$ 时，算法的划分结果已经非常接近了，没必要追求更低的 t 值。

如果 FOX 不考虑图的边权，算法的表现在表4.8中展示。由于 Fox 只计算三角形的数目，从结果上看，与带权的 Fox 算法相比，w_c 和 w_c/w_i 的值都变差了。

4.2　识别高影响力节点

4.2.1　应用背景

社交网络中的影响力最大化问题的研究有着十分重要的现实意义，它在市场营销、广告发布、舆情预警以及社会安定等方面皆有相关应用[133]。影响力最大化问题可概括为：给定一个社交网络图和一种特定的影响力传播模型，给定初始的传播节点个数，如何在图中确定这些初始的节点集合 (这些集合中的节点初始时是被激活的)，然后遵循

表 4.8　不同迭代停止条件下，Fox 在通话图上的表现。忽略边权

t	社区数目	时间（迭代次数）	社区大小	Density	w_c	w_i	w_c/w_i
0.15	437,477	11min (2)	12.25	0.461	24.62	12.37	1.99
0.10	430,381	17min (3)	13.42	0.436	24.37	12.25	1.99
0.05	425,972	20min (4)	13.91	0.427	24.25	12.23	1.98
0.02	423,463	24min (5)	14.11	0.424	24.21	12.22	1.98
0.01	422,134	29min (6)	14.20	0.422	24.19	12.23	1.98
0.001	421,161	43min (11)	14.27	0.422	24.17	12.22	1.97
0.0001	421,062	57min (12)	14.27	0.422	24.17	12.23	1.98
0.00001	421,031	75min (13)	14.28	0.422	24.17	12.22	1.97

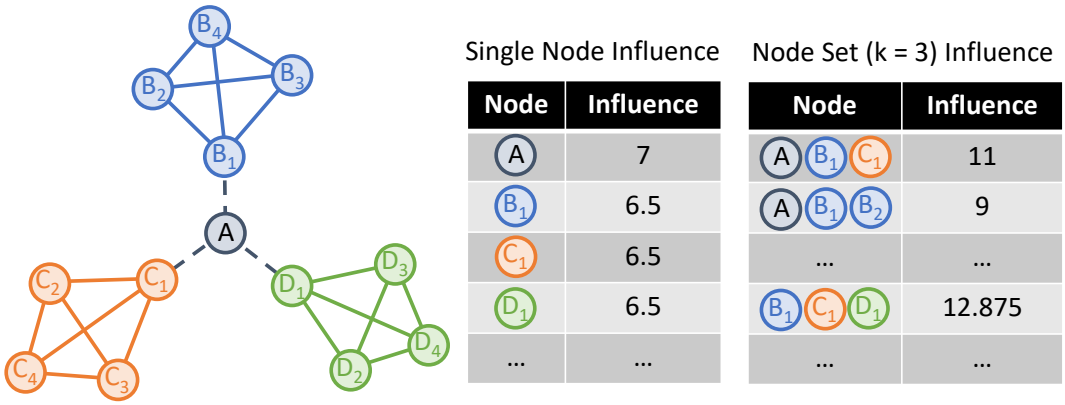


图 4.6　影响力最大化的传统方法所不适用的情形。

影响力节点的传播机制，从这些集合中的节点开始传播，使最终被影响的节点数目达到最多[133]，其形式化的表述如下：给定一个社交网络 $G(V,E)$，V 为节点集合，E 为边的集合，对于给定参数 k，k 是一个正整数，如何从图 G 中选择 k 个初始节点集合 A，满足 $|A|=k$ 且 $A \in V$，按照某种传播策略，由这 k 个初始的节点开始影响其它节点，并使最终被影响的节点数目达到最大。

传统的影响力最大化模型会计算每个节点的传播范围期望，选取期望最大的节点加入初始节点集合。然而这种策略在某些情况下是错误的。例如图 4.6 所示，传统的方法会将节点 A 率先加入初始节点集合。然而可以观察到有 A 节点的任何初始节点集都没有B_1, C_1, D_1 的影响力大。而B_1, C_1, D_1 正是图三个局部的中心点。这个例子说明，在处理影响力最大化任务时，目标节点影响的节点差异性也应被考虑。

本节认为信息传播模型有必要与节点差异性共同考虑，其原因可以总结为两个方面。首先，从实际应用的角度来看，很多应用场景需要找到既具备影响力，又有差异性的关键节点。例如在组建会议的程序委员会时，目标是召集在各个子领域有影响力的专家，而不单纯的是一个行业名人[134]。如果不考虑具体子领域，有可能会导致委员会不能给出公平周全的建议。第二点，从理论上讲，任何的传播模型都不能准确的刻

画真实的传播情况。原因在于传播模型的参数通常需要手动设定，而不是从数据中自动地学习出来。即便参数可以被学习出来，也会存在误差，尤其是在一些特殊的情景中，误差很大（例如预测社区之间的弱连接[135]）。如果选出的高影响力节点都属于同一个社区，这相当于把鸡蛋放在了一个篮子里，有很大的概率不能够达到最初的预期。相反，如果可以有意识地从多个社区中选取高影响力的节点，这会使得算法更加鲁棒。

本节所提出的模型尝试在考虑影响力最大化时，平衡好节点的权威性和差异性。其中，差异性定义为目标节点之间相似度的倒数。根据社会学的研究[88,136]，相似的节点有更大的概率在图中相连。因此可以利用社区发现的结果，并在此基础之上定义节点的差异性。在考虑权威性和差异性的平衡时，本节所提出的模型没有简单地选择加权求和的方式，而是运用了经济学中一些经典模型[137]。

4.2.2 节点的权威性和差异性定义

本文遵从相关研究中对于节点权威性的定义，根据图中信息传播的两个基本模型定义节点的权威性。

- IC 模型。当节点 u 在 t 时刻被激活，它可以以 p_{uv} 的概率在 $t+1$ 时刻激活其某个未被激活的邻居 v。如果 v 未被激活成功，那么在接下来的时间中都不能够被 u 激活。
- LT 模型。每个节点 v 都有一个取值范围在 [0,1] 之间的阈值 θ_v。每条边对应一个边权 b_{uv}。如果在 t 时刻，一个未被激活的节点 v 所收到的信息和大于阈值 $\sum_{v\text{'s active neighbor } u} b_{uv} \geq \theta_v$，则会在 $t+1$ 时刻被激活。

在图 $G=(V,E)$ 中，信息传播的发生通常从一些初始激活的节点 $S_0=S$ 出发，按照以上两种传播模型中的某一个，不断进行迭代。在这个过程中，将节点的权威性定义为传播范围的期望（比如说，以该节点为初始激活节点，IC 为传播模型，最终被激活的节点数目）：

$$\sigma(S)=\mathbb{E}[|S_N|]. \tag{4.15}$$

在信息检索领域，为了提高检索结果的多样性，在计算展示结果和用户查询项之间的相关性时，通常要把检索结果之间的相似度做为惩罚项加入[138]。类比这种处理方式，可以定义影响最大化问题中的节点的差异性：

$$d(S)=1-\frac{\sum_{u,v\in S,u\neq v}\mathtt{Sim}(u,v)}{Z}, \tag{4.16}$$

其中 $\mathtt{Sim}(u,v)$ 代表节点 u 和 v 之间的相似性。Z 是用作标准化的一个常数，以确保 $d(\cdot)$ 是非负的。

$\mathtt{Sim}(u,v)$ 可以用多种计算形式，比如基于语义信息、标签信息等等。通常来说，社

会学家认为图结构本身已经能够在一定程度上反应节点之间的相似度[79]。比如说，有相似爱好的人们在现实中更有可能是相识的朋友，有相同的研究兴趣的学者更有可能在相同的期刊或会议上发表文章。因此，本文在定义节点的差异性时主要基于图结构。

重叠社区发现算法通常将节点与社区之间的关系通过一个向量 $\boldsymbol{F_u} = [F_{u1}, ..., F_{uC}]$ 来表示，其中 F_{uc} 代表 u 节点属于社区 c 的概率。相似的表示方式对于离散的社区依旧成立。对于离散的社区划分来说，F_u 是一个 one-hot 向量。两个节点 u 和 v 之间的关系可以基于其从属社区的情况来计算。对于重叠社区来说：

$$p_C(u, v) = 1 - \exp(-\boldsymbol{F_u} \cdot \boldsymbol{F_v}^\top) \tag{4.17}$$

对于离散社区来说：

$$p_C(u, v) = (1 - \frac{1}{e}) \cdot \mathbf{1}_{c_u = c_v} \tag{4.18}$$

其中 c_u 指的是节点 u 所属的社区编号。

在 $p_C(u, v)$ 的基础之上，可以进一步定义 $\texttt{Sim}(u, v)$。当 $Z = k(k-1)$ 且 $|S| = k$ 时，有

$$d(S) = \frac{1}{k(k-1)} \sum_{u,v \in S, u \neq v} \Big(1 - \texttt{Sim}(u, v)\Big) = \frac{1}{k(k-1)} \sum_{u,v \in S, u \neq v} \texttt{dist}(u, v), \tag{4.19}$$

其中 $\texttt{dist}(u, v) = 1 - \texttt{Sim}(u, v)$ 是节点 u 和 v 之间的距离。此时，差异性可以被解释为 S 之中节点对之间的平均距离，这与文献[139,140]中是一致的。$\texttt{dist}(\cdot, \cdot)$ 可以被替换为 $p_C(u, v)$ 完成计算。同时也可以通过除了社区发现以外的其他方式来计算，如节点的向量表示等等。

4.2.3　权威性和差异性的平衡

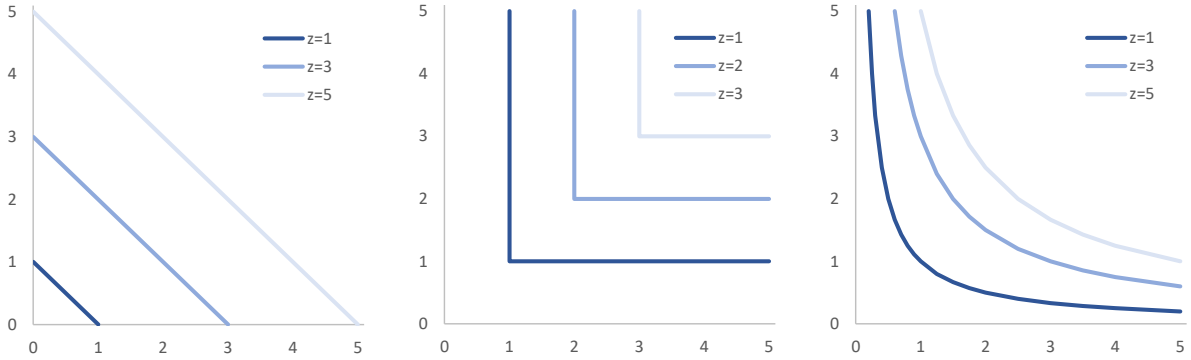
在保留原有节点影响力计算方式的情形下，设计了三种融合节点权威度 $\sigma(S)$ 和差异性 $d(S)$ 的方式，分别是：

Perfect Substitutes 方式 (PS) 当两个商品互为替代品时，顾客会以一个替代系数来在两个商品之间进行选择（比如百事可乐和可口可乐）。其数学形式是：

$$f_S(S) = \sigma(S) + c \cdot d(S). \tag{4.20}$$

其中 c 是一个非负的乘数。为了使得 $f_S(\cdot)$ 更加有意义，可以调整 c 使得 $c \cdot d(S)$ 和 $\sigma(S)$ 在相同的量级，例如 $c = |V|$。

Perfect Complements 方式 (PC) 当两个商品互为补充品时，比如一双鞋的左右脚，有更多的左脚鞋或者右脚鞋都对顾客没有益处。也就是说，如果节点的权威性和差异



(a) Perfect Substitutes (b) Perfect Complements (c) Cobb-Douglas

图 4.7 三种效用方程曲线。

性符合这种关系时，可以通过如下的方式进行组合：

$$f_C(S) = \min\{\sigma(S),\ c \cdot d(S)\}. \tag{4.21}$$

Cobb-Douglas 方式 (CD) 在经济学中，边际效益递减效应广泛存在：当保持其他因素不变，而单一增加某一元素，那么该元素给整个系统带来的单位收益不断减少。Cobb-Douglas 可以刻画这一过程：

$$f_D(S) = \sigma(S)^a \cdot d(S)^b. \tag{4.22}$$

4.2.4 差异性影响力最大化算法

有了三种效用方程，可以进一步形式化的定义差异性影响力最大化问题

定义 5 (DIM) *在图 $G = (V, E)$ 中，给定某效用方程 $f \in \{f_S, f_C, f_D\}$，目标 $\max_{|S| \le k} f(S)$.*

可以发现，差异性影响力最大化问题是影响力最大化模型的扩展，它的难度也是显而易见的。

定理 1 *当效用方程是 PS, PC 和 CD 时，DIM 问题都是 NP 难的。*

证明 1 *考虑一个 NP 难的节点覆盖问题：一个包含 n 个节点的图 $G = (V, E)$ 和一个整数 k，判断是否存在一个大小为 k 的节点子集 S，使得图 G 中每条边都至少有一个节点属于 S。这可以视作是 LT 影响力最大化问题的一个特例，差异性影响力最大化问题亦是 LT 影响力最大化问题的一个特例。针对该节点覆盖问题，假如存在一个符合要求的节点子集 S，那么可以认为以 S 为初始激活节点集合，$\sigma(S) = n$。*

由于 DIM 是 NP 难问题，可以将目标放在寻找近似最优点上面。根据文献，当目标方程具备单调性和次模性时，利用爬山贪心算法可以达到近似率为 $(1-1/e-\epsilon)$ 的次优点。然而，DIM 问题中的三个效用方程都具有次模性而无单调性。

定理 2 对任意的单调次模 $\sigma(\cdot)$ 和任意 $Sim(\cdot,\cdot)\in[0,1]$，三种效用函数 $f_S(S)$, $f_C(S)$ 和 $f_D(S)$ 都是非负的、次模的。

证明 2 $f_S(S)$ 是两个次模函数的和，因此也是次模的。

对 $f_C(S)$ 的次模性证明等同于证明存在 S,T 使得 $f_C(S)+f_C(T)\geq f_C(S\cup T)+f_C(S\cap T)$。

- 情形 *1*：如果 $f_C(S)=\sigma(S)$，$f_C(T)=\sigma(T)$，那么

$$\begin{aligned}
&f_C(S)+f_C(T)\\
=&\sigma(S)+\sigma(T)\\
\geq&\sigma(S\cup T)+\sigma(S\cap T)\\
\geq&\min\{\sigma(S\cup T),\beta\cdot d(S\cup T)\}+\min\{\sigma(S\cap T),\beta\cdot d(S\cap T)\}\\
=&f_C(S\cup T)+f_C(S\cap T)
\end{aligned}\tag{4.23}$$

- 情形 *2*：如果 $f_C(S)=\beta\cdot d(S)$，$f_C(T)=\beta\cdot d(T)$，那么证明的过程与情形 *1* 基本一致。
- 情形 *3*：如果 $f_C(S)=\sigma(S)$，$f_C(T)=\beta\cdot d(T)$，那么

$$\begin{aligned}
&f_C(S)+f_C(T)\\
=&\sigma(S)+\beta\cdot d(T)\\
\geq&\sigma(S\cup T)+\sigma(S\cap T)-\sigma(T)+\beta\cdot d(T)\\
\geq&\sigma(S\cup T)+\sigma(S\cap T)-\sigma(S\cap T)+\beta\cdot d(S\cap T)\\
=&\sigma(S\cup T)+\beta\cdot d(S\cap T)\\
\geq&f_C(S\cup T)+f_C(S\cap T)
\end{aligned}\tag{4.24}$$

- 情形 *4*：如果 $f_C(T)=\sigma(T)$，$f_C(S)=\beta\cdot d(S)$，那么证明的过程与情形 *3* 基本一致。

对 $f_D(S)$ 的次模性证明从 $a=b=1$ 的简单情形入手。首先给出如下标记：

$$\begin{aligned}
&\sigma(S\cup\{x\})-\sigma(S)=\Delta\sigma,\quad \sigma(T\cup\{x\})-\sigma(T)=\Delta\sigma-\epsilon_1\\
&d(S\cup\{x\})-d(S)=-\Delta d,\quad d(T\cup\{x\})-d(T)=-\Delta d-\epsilon_2,
\end{aligned}\tag{4.25}$$

其中 $\Delta d \geq 0, \epsilon_2 \geq 0, \Delta\sigma \geq \epsilon_1 \geq 0$，那么有

$$
\begin{aligned}
& f_D(T \cup \{x\}) - f_D(T) \\
=& d(T \cup \{x\}) \cdot \sigma(T \cup \{x\}) - d(T) \cdot \sigma(T) \\
=& (d(T) - \Delta d - \epsilon_2)\,(\sigma(T) + \Delta\sigma - \epsilon_1) - d(T) \cdot \sigma(T) \\
=& (\Delta\sigma - \epsilon_1)\, d(T) - (\Delta d + \epsilon_2)\, \sigma(T) - (\Delta\sigma - \epsilon_1)\,(\Delta d + \epsilon_2) \\
\leq& (\Delta\sigma - \epsilon_1)\, d(S) - (\Delta d + \epsilon_2)\, \sigma(S) - (\Delta\sigma - \epsilon_1)\,(\Delta d + \epsilon_2) \\
=& \Delta\sigma \cdot d(S) - \Delta d \cdot \sigma(S) - \Delta\sigma\Delta d - \epsilon_1(d(S) - \Delta d) - \epsilon_2\sigma(S) - \epsilon_2\,(\Delta\sigma - \epsilon_1) \\
\leq& \Delta\sigma \cdot d(S) - \Delta d \cdot \sigma(S) - \Delta\sigma\Delta d \\
=& f_D(S \cup \{x\}) - f_D(S).
\end{aligned}
\tag{4.26}
$$

针对其他情形，只需证明 $\sigma(\cdot)^a$ 是非负的、单增的、次模的，同时 $d(\cdot)^b$ 是非负的、单减的、次模的，然后沿用上面的方法证明二者的乘积是次模的。对于 $\sigma(S)^a$ 来说，其非负性和单调性是显然的。$\sigma(S)$ 是单调的、次模的，$x^a (0 \leq a \leq 1)$ 是非降的凹函数，它们的组合 $\sigma(S)^a$ 因此也是次模的。对于 $d(S)^b$ 的证明是类似的。

根据定理 2，DIM 是一个带尺寸约束的（对初始激活节点数量的约束）非单调的次模最优化问题。在文献[141]中提出的 RandomGreedy 算法可以适用于 DIM 问题。RandomGreedy 是爬山贪心法的一种扩展形式。在每一轮迭代中，算法不是挑选一个当前最优的节点，而是挑选前 k 个带来最大边际收益的节点，并随机从中挑选一个加入节点集。因此，使用 RandomGreedy 的方法来构造初始点集，算法形式化的表述如算法2所示。

Algorithm 2 `RandomGreedy`(k, f)

```
initialize S_0 = ∅
for i = 1 to k do
    Let M_i ⊆ V − S_{i−1} be the subset of size k maximizing Σ_{v∈M_i} f(S_{i−1} ∪ {v}) − f(S_{i−1})
    Randomly select u from M_i
    S_i = S_{i−1} ∪ {u}
end for
output S_0
```

定理 3 (Buchbinder et al.[141]) 令 $f(\cdot)$ 为一个非负的次模函数。对于问题 $\max_{|S| \leq k} f(S)$，RandomGreedy 可以保证 $\mathbb{E}[f(S)] \geq \frac{1}{e} \cdot f(S^*)$，其中 S^* 是问题的最优点。

证明 3 在文献[141]中，已给出了该定理的证明。

根据定理 2和3，以 PS, PC 或者 CD 为效用方程的 DIM 问题可以达到近似率为 $(1/e-\epsilon)$ 的次最优点。其中 "ϵ'' 是在计算 $\sigma(\cdot)$ 时出现的误差。通过蒙特卡洛法可以尽可能地将该误差减小。

4.2.5　实验测评

4.2.5.1　实验准备

实验部分试图验证两个关键问题：

1. 和其他的关键节点挖掘算法相比，本节的方法是否能够取得更高的效用值？
2. 抛去效用函数的定义不谈，本节的算法是否能在保持传播影响范围的前提下，有效提升传播结果的多样性？

对比算法包括：

1. LINE[39]是一个图表示学习方法，为每个节点生成一个表示向量。利用 k-means 法对学习到的表示向量进行聚类，并选择每一类的中心节点做为种子节点。
2. PageRank[142]选取 PageRank 高的节点做为种子节点。
3. GenDeR[143]是一个多样性的排序算法，支持任意的排序方程和相似度方程。在实验中选取 PageRank 为排序方程，$(1-1/e)\mathbf{1}_{c_i=c_j}$ 为相似度方程。
4. IMGreedy 是一个传统影响力最大化算法的扩展算法[133]。
5. RandomGreedy 是本文提出的方法。对于效用函数 CD，设定 $a=b=1$；对于效用函数 PS 和 PC，设定 $c=0.05|V|$，$Z=k(k-1)$。

4.2.5.2　效用值最大化

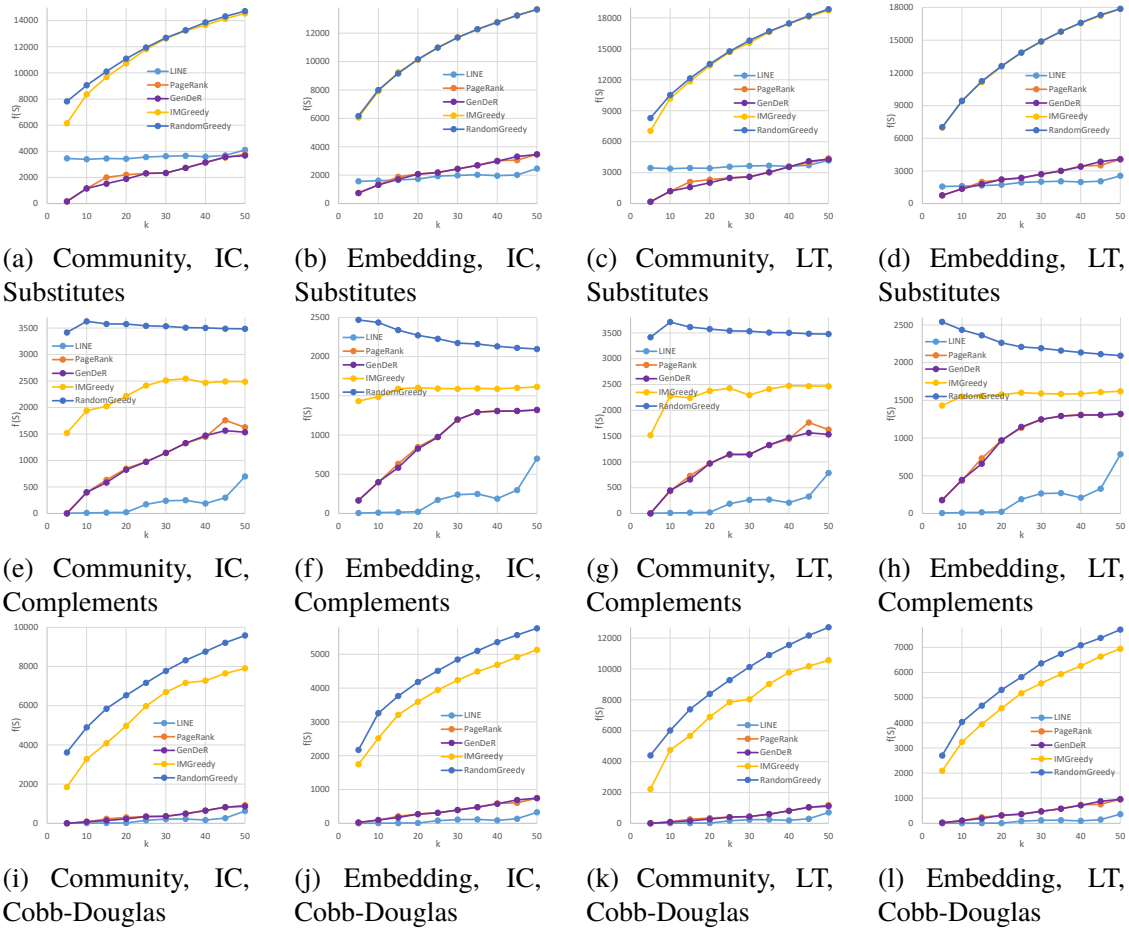
本实验选择两个图数据集。

1. Epinions 来自于顾客评价网站 Epinions.com，是一个代表用户之间信任关系的有向图，包含 75,879 个节点，508,837 条边。
2. NetHept 来自于学术论文网站 arXiv.org，是一个代表学者之间合作关系的无向图，包含 15,233 个节点和 58,991 条边。

在图4.8中展示了几个算法在 Epinions 数据集上的表现。当效用方程是 PC 和 CD 时，RandomGreedy 与其他算法相比有显著提升。当效用方程是 PS 时，RandomGreedy 仍然表现最好，不过与第二名 IMGreedy 的差距较小。这一观察与定理 s 3 是一致的。

4.2.5.3　传播多样性

更高的效用值在一定程度上代表算法性能的优秀，但并不意味着能够带来令人满意的结果。在这个实验中，借鉴已有文献中的评测方法[144,145]，尝试从另一个角度来判断结果是否展现出良好的多样性。在这个实验中，选用了一个演员网络 Imdb，以演员



(a) Community, IC, Substitutes (b) Embedding, IC, Substitutes (c) Community, LT, Substitutes (d) Embedding, LT, Substitutes

(e) Community, IC, Complements (f) Embedding, IC, Complements (g) Community, LT, Complements (h) Embedding, LT, Complements

(i) Community, IC, Cobb-Douglas (j) Embedding, IC, Cobb-Douglas (k) Community, LT, Cobb-Douglas (l) Embedding, LT, Cobb-Douglas

图 4.8　在 Epinions 数据集中，不同融合方式对最终影响力的影响。

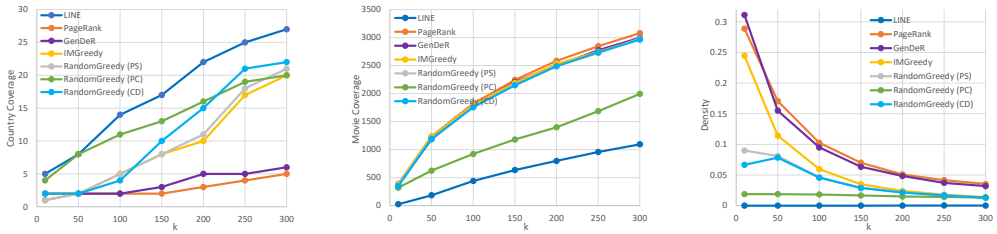
为节点，共同出演关系为边，同时每个演员还对应着其国籍信息。这个图涉及 5,044 部影片，6,271 个演员，以及 15 ,060 条边。

在衡量多样性上，计算两个指标：密度和覆盖度。其中密度指的是 k 个节点所形成的导出子图中边存在的概率。

$$\texttt{Density}(S) = \frac{\sum_{u \in S} \sum_{v \in S, u \neq v} \mathbf{1}_{(u,v) \in E}}{|S| \times (|S| - 1)}. \tag{4.27}$$

覆盖度指的是所选演员的国籍覆盖度和影片覆盖度。

评测结果如图4.9所示。可以认为国籍覆盖度和密度两个指标更能有效的反应节点的多样性，而影评覆盖度更多反应的是节点的权威性。在图4.9(b) 中，LINE 和 RandomGreedy (PC) 表现不佳；在图4.9(a) 和 4.9(c) 中，LINE 和 RandomGreedy (PC) 又成为了最好的两个。从两个角度来解释这一现象：首先，如果希望在尽可能保持权威性的基础上提升多样性，RandomGreedy (CD) 是最好的选择。其次，三种效用方程其实各有各的优势。PS 更倾向于权威性，PC 强调多样性，而 CD 像是在两者之间找到一个平衡点。



(a) 国籍覆盖度 (数值越高越好)　(b) 影片覆盖度 (数值越高越好)　(c) 密度 (数值越低越好)

图 4.9　IMDB 数据集结果的差异性测评。

4.3　本章小结

本章重点关注邻域解构策略。首先提出体现邻域解构的大规模重叠社区发现算法— Fox。Fox 将“图中节点的划归问题”分解为“目标节点与其邻居节点是否同属一个社区”的子问题。每个子问题仅需要关注目标节点和其邻域组成的导出子图。在本章中，尝试从博弈论（潜博弈）的角度解释 Fox 算法的意义，并且进一步证明其收敛性。为了加快算法的处理速度，本章亦改进了 Fox 算法中关于节点之间紧密程度的计算步骤，在保证算法效果的前提下有效提升了算法速度。截止到论文发表，Fox 算法是唯一一个可以处理 2200 万节点，1 亿条边（实验中的 Google+ 数据集）的重叠社区发现算法。该工作已经发表在 ICDM 2016 上。

随后，本文将大规模社区发现算法的结果应用于影响力最大化任务之上。传统的影响力最大化任务仅仅关注节点的权威性、影响力，目标是找到影响力最大的节点集合。本文在此基础之上引入了节点集差异性的概念，试图找到影响力大并且节点之间差异性较大的节点集合。这种改进方式更符合现实任务的需求。

第五章　图挖掘中路径解构策略的运用

本章介绍路径解构策略在图挖掘任务中的运用。在社交、生物、金融等多个领域之中，图上的传递性（transitivity）、集群性（clusterability）、互惠性（reciprocity）等广泛存在，揭示了图的拓扑结构与节点标签节点属性的关系。正因如此，大量的图挖掘任务涉及到图中节点关系的建模，以推断节点和边上的信息。进一步讲，常见的节点关系建模包括节点之间的最短路径、连接紧密程度、中心度的计算等等，其本质是在不同维度上刻画节点之间的邻近关系。然而当图的规模不断增大，这些问题的计算复杂度成为了其主要的限制因素。本章所关注的路径解构的策略可以在一定程度上解决这一问题。

本章的第一部分关于图的表示方法，关注节点的连续表示，提出一个体现路径解构的图神经网络算法。图神经网络的目标是对以图为形式组织在一起的复杂数据进行建模，从而支持节点标签预测、关系预测等任务。本文提出的算法 CNE，从模型功能上讲，将刻画图中一对节点的关系解构为计算随机游走路径中，两个节点依次出现的概率；从模型处理的数据上讲，将原始图解构为包含两个节点的游走路径。CNE 算法综合多条游走路径，少数边的缺失不会对节点之间的关系的计算造成显著影响，缓解了图结构信息缺失的问题；产生随机游走路径的代价远低于传统的临近度计算，解决了临近度计算复杂的问题。

本章的第二部分关于图挖掘对具体应用，提出一个新的大规模图中节点中心度计算指标。以中介数、特征向量中心度等为代表的节点全局中心度指标都受限于计算复杂度过高，难以适用于大规模图。本文提出的新中心度 Node Conductance 基于体现路径解构的图表示学习算法，对于大规模图可以快速完成计算。Node Conductance 的计算同样基于随机游走，解构策略的运用与 CNE 算法一致。其独特之处在于它所考虑的不是两个节点出现在随机游走同一窗口的概率，而是目标节点和自身同时出现在窗口的概率。

5.1　基于路径解构的图神经网络 CNE

5.1.1　研究背景

最近，图神经网络受到研究者们广泛的关注。被视为卷积神经网络在图结构数据上的拓展，图神经网络可以处理包含复杂图结构、节点属性、边属性的数据，支持完成图分类、节点分类、边预测等一系列相关下游任务。由于下游任务通常与节点密切

相关，现有的图神经网络的目标都可以被解释为学习到节点编号与节点表示向量之间的映射关系。然而，本节认为图神经网络研究中存在的众多挑战点都与这一设计模式有关。

首先节点编号并不具有泛化性，也就是说，大部分已有工作需要节点出现在训练阶段，否则无法给出节点的向量表示。然而现实生活中的图都是在不断演化中的，比如 Amazon 中的新用户和新商品，Youtube 上的新视频等等。不断变化的图需要图表示学习算法可以快速应对。倘若图结构更新后，重新计算节点的向量表示，将耗费大量的计算资源。有一些增量式[146] 或者直推式[57,76] 的方法可以推测出新节点的向量表示。但他们需要新节点与原始图之间存在连结关系。真实情境中，新产生的节点往往是孤立的，例如推荐系统中的冷启动商品，节点与图不存在任何连边。在这种状况下，已有的图神经网络无能为力。

另一个难点特别针对异构图而言。节点的编号并不是内生的携带节点类型信息。因此，试图利用模型直接学习到节点编号和节点类型的映射关系是不合理的。已有的工作[147–149]无法自然的表示多种类型的节点或边。它们通常将一个异构图拆解成为多个同构图，并在多个图上分别做向量表示，再花费大量计算资源将其结果对齐。

最后一个难点在于图神经网络模型对于图拓扑结构的敏感度和鲁棒性之间的平衡。一个理想的模型能够尽可能地保留输入的图拓扑信息，同时对微小的结构变动保持鲁棒。然而，当边被一对节点的编号所表示时，节点编号实际上携带的可资利用的信息几乎没有，对于边是否存在的分析也受制于此。因此，鲁棒性鲜被研究者们关注[150]。基于随机游走的方法对于结构的微小变化是鲁棒的，但这种鲁棒性是源于采样方法的随机性，而不是基于理性的推理。做为对比，基于融合邻域属性的方法[57,76] 由于完全依赖图拓扑结构，其结果会被结构的变化严重影响[151,152]。

这两个难点实际上源于一个原因，已有的图表示学习以节点本身作为基本单位，针对每个节点作出向量的优化。在这种状况下，算法无法推测未出现的节点，也无法分别处理多个类别的节点。而这两个难点的对策也在于此，可以将节点分解为它们所携带的节点特征。在真实的图中，图结构通常不是独自存在的。它们还带有各种各样丰富的节点特征、边特征等。

本节认为节点的向量表示可以由它们的特征融合得到。这一想法是受“复合性原理”[153] 启发得到。“复合性原理”指的是一个复杂表达式的意义是由其各组成部分的意义以及用以结合它们的规则来决定的。另一个相似的概念是“组合泛化”，它指的是可以通过已有模块的组合作出推断。这在语言学中得到广泛的应用，利用有限的词汇可以组合得到无穷无尽的语句含义。在这里借鉴这一思想，希望对图神经网络作出改进。与已有算法的普遍做法不同，CNE 没有为每个节点学习出一个表示向量。相反，CNE 学习一个融合节点属性的方法，通过把节点所携带的属性融合，得到相应节点的

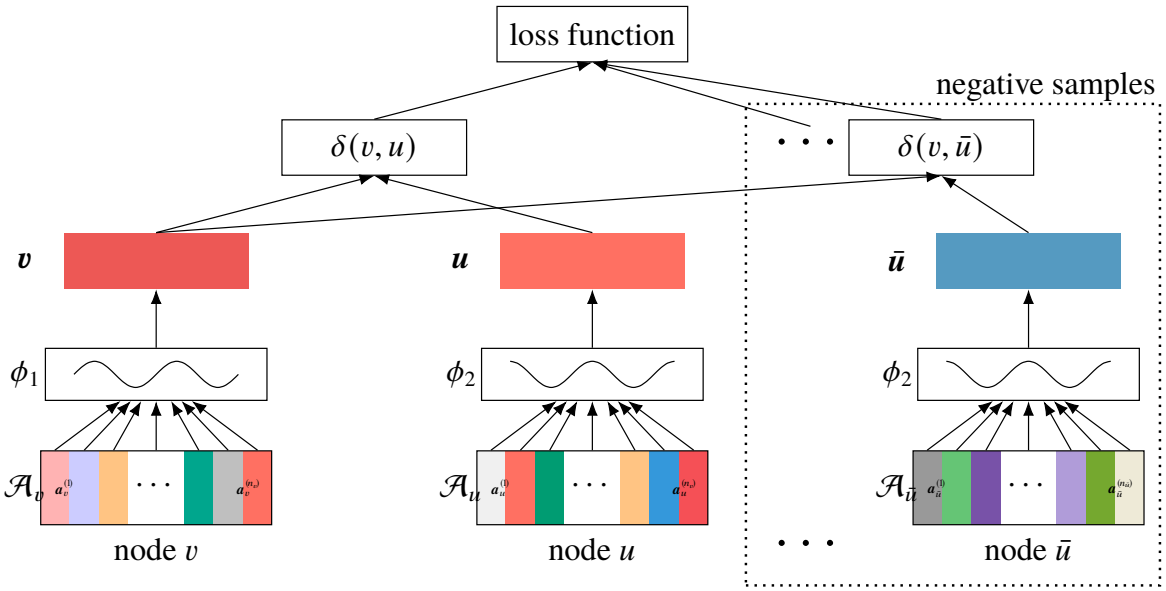


图 5.1　组合式图神经网络框架。对于一个正样本：目标节点 v 和其邻居 u，算法随机采样 K 个非邻居节点 $\bar{u}$。优化的目标是根据节点的表示向量将正样本和负样本区分开，其中表示向量是由节点的属性融合得到。

向量表示。当有新的节点出现时，CNE 可以将节点的特征加以融合，推测出该点的向量表示。实际上，节点向量表示是 CNE 模型中的一个中间计算量。

与传统的图神经网络模型相比，本章中提出的方法 CNE 原生地具备如下优势：

1. **能够推断未在训练阶段出现的节点的表示向量。**一旦 CNE 训练完成，模型可以基于节点所携带的属性信息推断出其表示向量。
2. **能够轻松地处理异构图。**不同的节点类型对应不同的节点属性和组合方式。也就是说，节点的类型可以很自然地被模型捕捉并建模。
3. **能够识别噪音边。**CNE 以公用的节点属性和组合方式为基础，建模网络拓扑结构。这也可以被视为是一个强有力的正则化方式。CNE 可以自然地忽视反常的边。反常的边指的是相连的两个节点所带的特征很罕见地共同出现，例如一个皇家马德里队的球迷点击了巴塞罗那队的球衣。

5.1.2　模型描述

正如图5.1展示的那样，CNE 包含两个关键的部分：

1. 融合节点属性的组合方程;
2. 基于图邻接结构的损失函数，用来学习属性的向量表示和融合方式。

在这里，为了不失一般性，以文本作为节点特征来介绍本节提出的模型框架。其他类型的节点特征也可以用不同的融合方式来处理。

5.1.2.1 特征融合

如图5.1所示，对于节点 v_i 向量表示的融合操作十分简单直接。通过融合方程 ϕ，v_i 的属性得到融合 $\mathcal{A}_i$=$[\boldsymbol{a}_i^{(1)}, \ldots, \boldsymbol{a}_i^{(n_i)}]$:

$$\boldsymbol{v}_i = \phi(\mathcal{A}_i) = \phi\left(\boldsymbol{a}_i^{(1)}, \ldots, \boldsymbol{a}_i^{(n_i)}\right)$$

其中 $\boldsymbol{a}_i^{(j)} \in \mathbb{R}^d$ 是节点的表示向量，v_i 的属性是 $a_i^{(j)}$，d 是 $\boldsymbol{a}_i^{(j)}$ 的维度，n_i 是 v_i 的特征个数。可以将融合方程 ϕ 用一个神经网络替代。在接下来的论述中也会将融合方程称作一个编码器。值得特别说明的是，特征 a 和它对应的表示向量 $\boldsymbol{a}$ 是所有节点共享的。这样，当有新的节点出现时，可以利用已经学习到的特征向量表示融合得到新节点的向量表示。

5.1.2.2 融合方式（编码器）

CNE 是一个通用的图表示学习框架。此处的通用性指的是 CNE 可以处理多种的节点属性的类型和融合方式。例如，对于文本类的节点属性（论文摘要），图片类型的节点属性（商品示意图），CNE 都可以对他们进行建模。针对不同的属性类型，可以设计不同类型的编码器，包括简单的拼接、取均值、求和操作，或者复杂的 GRU[154]，CNN[155]。

在这里主要关注文本类型的节点属性。用一个以 GRU 为单元的 RNN 编码器处理节点的文本属性。GRU 被证明和 LSTM 有着相当的能力，然而计算复杂度却相对较低[154]。

使用 RNN 的最后一个隐向量 $\boldsymbol{h}_{v_i}^{(n_i)}$ 当作节点的向量表示 $\mathcal{A}_i$。在每一步 t 中，GRU 的计算过程是（在这里隐去了下标）：

$$\begin{aligned}
\boldsymbol{r}^{(t)} &= \sigma\left(\boldsymbol{W}_r \boldsymbol{a}^{(t)} + \boldsymbol{U}_r \boldsymbol{h}^{(t-1)}\right) \\
\boldsymbol{z}^{(t)} &= \sigma\left(\boldsymbol{W}_z \boldsymbol{a}^{(t)} + \boldsymbol{U}_z \boldsymbol{h}^{(t-1)}\right) \\
\tilde{\boldsymbol{h}}^{(t)} &= \tanh\left(\boldsymbol{W}\boldsymbol{a}^{(t)} + \boldsymbol{U}(\boldsymbol{r}^{(t)} \odot \boldsymbol{h}^{(t-1)})\right) \\
\boldsymbol{h}^{(t)} &= \left(1 - \boldsymbol{z}^{(t)}\right) \odot \boldsymbol{h}^{(t-1)} + \boldsymbol{z}^{(t)} \odot \tilde{\boldsymbol{h}}^{(t)}
\end{aligned} \tag{5.1}$$

其中 σ 是 sigmoid 方程，$\odot$ 是按位乘法，$\boldsymbol{r}^{(t)}$ 是重制门，$\boldsymbol{z}^{(t)}$ 是更新门，所有的非线性计算都是按位计算的，$\boldsymbol{W}_r, \boldsymbol{W}$, 和 $\boldsymbol{U}$ 在这里都是可以学习得到的参数。在这里使用了 GRU，实际上任何结构都可以在这里被使用，只要梯度可以被回传到属性的向量表示即可。

5.1.2.3　学习 CNE 的模型参数

为了能学习到有效的属性向量表示，CNE 设计了一个基于图结构的损失函数，参数通过随机梯度下降得到优化。损失函数鼓励临近的节点有着相似的节点向量表示，不临近的节点有着较远的向量距离。CNE 是一个端到端的双塔模型。对于节点 v 和它的邻居 $u \in \mathcal{N}(v)$，CNE 设定一个 max-margin 损失函数：[156]：

$$\mathcal{L}(v,u) = \sum_{k=1}^{K} \max\big(0, m - \delta(v,u) + \delta(v,\bar{u}_k)\big) \tag{5.2}$$

其中 $\bar{u}_k$ 是从全局点集 $\mathcal{V}$ 中随机选取的负样例，K 是负样例的个数；m 是正负样本之间的差距，通常会设置为 1；δ 是衡量两个向量距离的函数：

$$\delta(v,u) = \cos(\boldsymbol{v},\boldsymbol{u}) = \cos\Big(\phi_1(\mathcal{A}_v), \phi_2(\mathcal{A}_u)\Big) \tag{5.3}$$

其中 ϕ_1 和 ϕ_2 分别是节点 v 和 u 的编码器。

在已有工作中[10,39,45]，$\boldsymbol{v}$ (和 $\boldsymbol{u}$) 是通过查表的方式根据节点编号查询得到的。而在 CNE 模型中，$\boldsymbol{v}$ (和 $\boldsymbol{u}$) 是通过组合节点携带的特征的方式得到的。这种组合式的向量生成方式使得模型在训练的过程中，节点属性和图拓扑结构信息得到了充分的交互，模型因而得以捕捉到基于图结构的属性相似性。

直观地说，max-margin 函数的目标是使得一对正样本的打分与一对负样本的打分相差至少为 m。实际上，可以根据不同的目标和应用背景设计不同的类型的损失函数。CNE 同样可以以监督学习的方式进行训练，只需把当前无监督的损失函数替换为带有节点标签的损失函数即可。

5.1.2.4　邻域的定义

在上文提到的损失函数中，邻域是一个非常重要的概念。在本节中，沿袭经典算法 DeepWalk[10]的高效做法，基于随机游走定义邻域。以每个节点为起点，生成多条长度为 l 的随机游走路径。随后，节点 v 的邻居被定义为：随机游走路径中，那些与该节点共同出现在大小为 w 的窗口中的节点。这种采样方式在节点的向量表示与图拓扑结构之间建立了联系。

5.1.3　模型变种

上文中介绍了 CNE 模型的框架。接下来将讨论如何在真实复杂的图中利用 CNE 模型。

1. **有向图。**边的方向需要被模型捕捉。为了达成这个目的，需要为边的起点和终点设计参数不共享的编码器，ϕ_1 和 ϕ_2，这样由于 $\phi_1(\mathcal{A}_v) \neq \phi_2(\mathcal{A}_v)$，$\delta(v,u) \neq \delta(u,v)$，

边的方向得以区分。

2. **节点异构图**。通常不同种类的节点对应着不同的节点特征。比如，在推荐问题中，商品的种种属性信息与用户的基本信息是完全不同的。可以为不同类型的节点设计不同的编码器。比如，利用 RNN 去编码文本特征，用 CNN 去编码图片的信息，用更复杂的图结构建模更复杂的节点特征。
3. **边异构图**。类似于有向图的做法，CNE 可以为不同类型的边设计不同的编码器，但编码器共享节点属性的编码信息。例如，在一个包含关注和转发关系的有向社交网络中，可以设计 4 个编码器，两个用来建模关注关系，另外两个建模转发关系。在这种情形下，CNE 更像是一个多任务模型。

5.1.4 实验评测

在本部分实验中，将测试算法在四个不同情境下的边预测任务，以证明算法在同构图和异构图上的易用性和优秀表现。

5.1.4.1 实验准备

为了证明 CNE 算法的有效性，将其与几个优秀的基线算法进行对比：

1. **SGNS**[157]: 该算法利用 SGNS 算法习得节点标签的词向量，在实验中，每个节点的向量表示为所对应的词向量之和。
2. **DeepWalk**[10]: 该算法以随机游走为采样方法的 Skip-gram 模型，为每个节点习得一个仅与结构相关的表示向量。
3. **CANE**[158]: 该算法基于注意力机制和节点上的文本信息，为每个节点习得一个与邻域相关的表示向量。
4. **TriDNR**[159]: 该算法综合考虑图结构、节点和节点属性、节点属性和节点标签之间的关系，习得一个综合三方面信息的表示向量。
5. **GraphSAGE**[57]: 该算法是一个典型的直推式方法。它综合每个目标节点的邻域节点为目标节点生成表示向量。

在本实验中，所有的基线算法都是用其作者发布的代码实现。为了公平对比，为所有被测算法设计相同的超参数（在验证集上调节到最优）。节点表示向量的维度为 512。对于那些利用文本信息的算法，建立了一个 40000 词的词表，并重新训练词向量。对于 GraphSAGE 算法来说，聚合操作通过求取均值来实现，所利用的输入节点属性为 SGNS 训练得到的词向量。对于基于随机游走的算法 (包括 DeepWalk, TriDNR, 和 CNE)，本实验将随机游走的长度设为 l=20, 窗口大小设为 w=2, 对于每个正样本来说，随机负采样的数目为 K=4。CNE 中涉及 GRU 编码文本信息，将词向量维度设为 256，GRU 隐层的维度为 512，注意力机制未使用。CNE 使用 Adam 优化方法，初始学习率为 0.0008。

表 5.1　不同测试集大小的召回值 (任务 1)。

	10%			30%			50%			70%			90%		
Method	R@10	R@50	R@100	R@10	R@50	R@100	R@10	R@50	R@100	R@10	R@50	R@100	R@10	R@50	R@100
SGNS	0.054	0.108	0.134	0.056	0.108	0.136	0.055	0.109	0.136	0.055	0.111	0.138	0.056	0.111	0.137
DeepWalk	0.103	0.270	0.389	0.102	0.296	0.430	0.112	0.319	0.456	0.114	0.324	0.462	0.116	0.329	0.469
TriDNR	0.091	0.194	0.243	0.095	0.217	0.278	0.102	0.231	0.299	0.101	0.238	0.309	0.104	0.247	0.329
CANE	0.140	0.354	0.473	0.138	0.355	0.478	0.144	0.364	0.484	0.138	0.352	0.477	0.139	0.358	0.483
GraphSAGE	0.088	0.230	0.308	0.101	0.290	0.396	0.113	0.320	0.437	0.112	0.323	0.446	0.103	0.314	0.436
CNE	**0.154**	**0.422**	**0.547**	**0.152**	**0.422**	**0.551**	**0.157**	**0.435**	**0.564**	**0.162**	**0.452**	**0.585**	**0.176**	**0.482**	**0.622**

5.1.4.2　评测任务

图的结构是在不断变化更迭的，因此边预测是图挖掘中一个经典的任务。电商平台中的商品推荐，社交网络中的可能认识的人都是边预测的典型应用场景。以淘宝平台举例，每天有上百万的新商品上架，同时也有上百万的旧商品下架。对于旧商品，可以通过协同过滤这一经典算法，挖掘出用户潜在的喜好与特性。那么如何将新商品推荐给用户呢？新商品不会与任何用户产生交互行为，在协同过滤矩阵中很难计算其与用户的相关性。

实验主要围绕边预测任务展开，关注不同情形下的边预测成功率。实验中随机移除一些图中的边，并用处理过后的图训练各个算法。在测试中，从图中随机抽取 1000 个节点，根据这 1000 个节点的表示向量预测先前被移除的边。和以往的边预测实验设置不同，本实验并没有将边预测的任务转化为一个二分类的任务，而是与论文[40]一致，以一种更贴合实际应用场景的方式来进行实验。根据习得的表示向量，可以轻松地估计出节点之间的相似性。相似性高的节点对意味着两者之间有更高的概率相连。这样，可以利用 Precision@k 和 Recall@k 来衡量边预测的效果：

$$\begin{aligned} \text{Precision@}k &= \frac{\#(\text{real neighbors} \cap \text{top } k \text{ candidates})}{k} \\ \text{Recall@}k &= \frac{\#(\text{real neighbors} \cap \text{top } k \text{ candidates})}{\#(\text{real neighbors})}. \end{aligned} \tag{5.4}$$

实验为每个节点都计算 Precision@k 和 Recall@k 值，并在下面的实验分析中汇报所有参与测试节点的平均值。值得注意的是，两个指标的计算都需要从节点候选集中截取最相似的 k 个节点，将节点候选集设定为整个图的节点全集，而不是其他论文中一个特殊的测试节点子集。本文的实验设计更加困难，同时也更符合算法运用的真实场景。

5.1.4.3　任务一：同构图上的边预测问题

本实验使用 Amazon 的 Baby 类目作为数据集，该数据集包含 71317 条商品的信息。利用用户的共同浏览关系，构造了一个商品的同构图。该图包含 47185 个节点，1166828 条边。隐去的边的比例从 10% 到 90%。本实验将商品的名称当作节点属性。如表5.1和表5.2所示，相较于对比算法，CNE 可以取得更好的得分。DeepWalk 和 CANE

表 5.2　共同浏览网络中边预测任务的 Precision@k (任务 1)。

	10%			30%			50%			70%			90%		
算法	P@10	P@50	P@100	P@10	P@50	P@100	P@10	P@50	P@100	P@10	P@50	P@100	P@10	P@50	P@100
SGNS	0.131	0.056	0.036	0.132	0.056	0.036	0.131	0.057	0.036	0.133	0.058	0.037	0.134	0.058	0.037
DeepWalk	0.208	0.135	0.105	0.220	0.153	0.117	0.237	0.162	0.123	0.237	0.163	0.123	0.244	0.166	0.125
TriDNR	0.190	0.094	0.062	0.203	0.108	0.072	0.215	0.117	0.078	0.213	0.120	0.082	0.219	0.126	0.086
CANE	0.315	0.187	0.130	0.320	0.190	0.133	0.328	0.193	0.133	0.315	0.185	0.130	0.316	0.187	0.131
GraphSAGE	0.216	0.125	0.086	0.252	0.159	0.111	0.263	0.170	0.120	0.261	0.171	0.123	0.236	0.165	0.120
CNE	**0.374**	**0.231**	**0.155**	**0.369**	**0.233**	**0.157**	**0.389**	**0.241**	**0.161**	**0.390**	**0.249**	**0.166**	**0.414**	**0.262**	**0.174**

表 5.3　共同浏览网络中冷启动节点边预测的 Precision@k (任务 2)。

	10%			30%			50%		
算法	P@10	P@50	P@100	P@10	P@50	P@100	P@10	P@50	P@100
SGNS	0.016	0.005	0.003	0.016	0.005	0.003	0.016	0.005	0.003
TriDNR	0.019	0.005	0.003	0.021	0.006	0.004	0.030	0.009	0.005
CNE	**0.034**	**0.013**	**0.008**	**0.039**	**0.014**	**0.009**	**0.042**	**0.015**	**0.009**

需要测试的节点也出现在训练的过程中。为了公平比较，规定测试的节点都出现在训练过程中。

拓扑结构的重要性　DeepWalk 在实验中是一个很强的对比算法。这说明，结构本身对边预测提供了很多的信息。SGNS 单纯的基于节点的特征，从词语的共现次数中捕捉词语的相似性。CNE 和 SGNS 之间的差距证明了图的拓扑结构有效的提升了词语相似性的学习。这一结论与文献[160]中的结论一致。

同时训练拓扑结构和词语表示的重要性　虽然节点属性可以很好地消除边稀疏的问题，但节点属性也需要以一种合适的方式加以利用。与 DeepWalk 相比，CANE 和 CNE 利用了属性信息，取得了较好的分数，尤其当边比较稀疏的时候。然而不同的算法利用文本信息的方式不同，TriDNR 和 GraphSAGE 同样利用了文本信息，但是效果却变差了。CNE 以一种端到端的方式同时更新节点的向量表示和文本的向量表示，在边预测这一任务上可以取得更好的效果。

5.1.4.4　任务二：针对冷启动节点的边预测

除了对于测试节点的选择有所不同，该实验与任务一基本一致。在这个任务中重点考察那些没有出现在训练集中的节点。GraphSAGE，DeepWalk，和 CANE 无法完成这个任务。因此只在表5.3中展示了其他对比算法的表现。在这个任务中，训练边的比例在 10% 到 50% 之间，如果保留更多比例的边就无法产生足够多的测试点了。

表5.3说明，CNE 在多数情况下仍然是最优秀的表示算法。TriDNR 分别建模了节点的结构向量和文本向量。当新节点的结构信息不存在时，单纯依赖文本向量极大地削弱了 TriDNR 的表示能力。相反，CNE 学习到的文本向量表示在一定程度上已经包

表 5.4　浏览后购买网络中边预测任务的 Precision@k (任务 3)。

	20%			40%			60%			80%		
算法	P@10	P@50	P@100	P@10	P@50	P@100	P@10	P@50	P@100	P@10	P@50	P@100
SGNS	0.033	0.011	0.007	0.034	0.011	0.007	0.035	0.012	0.007	0.036	0.012	0.007
DeepWalk	0.093	0.032	0.019	0.099	0.038	0.023	0.095	0.037	0.023	0.090	0.036	0.022
CANE	0.080	0.029	0.017	0.092	0.033	0.019	0.091	0.033	0.019	0.091	0.033	0.019
TriDNR	0.065	0.022	0.013	0.068	0.024	0.014	0.078	0.028	0.016	0.078	0.029	0.017
GraphSAGE	0.056	0.020	0.012	0.063	0.024	0.014	0.067	0.026	0.016	0.068	0.027	0.016
CNE	0.081	0.029	0.019	0.085	0.034	0.022	0.083	0.033	0.024	0.082	0.036	0.030
CNE$_{MUL}$	**0.120**	**0.040**	**0.022**	**0.128**	**0.041**	**0.022**	**0.134**	**0.047**	**0.027**	**0.136**	**0.054**	**0.033**

含了图的结构信息，因此相较 TriDNR 的文本向量会有更丰富的信息量。因此 CNE 会比 TriDNR，SGNS 有着更亮眼的表现。值得注意的是，任务二比任务一的难度高很多，由于冷启动节点在图中的连边很少，因此边预测命中的概率也大大降低。

5.1.4.5　任务三：边异构图的边预测

本任务的设置为了证明 CNE 可以适用于多关系的图。仍然使用 Amazon Baby 类目的数据，考察其中的共同浏览关系和浏览后购买关系。共同浏览网络包含 47185 个节点，1166828 条边，浏览后购买包含 44078 个节点，111473 条边。浏览后购买网络的边预测结果在表5.4中展示

浏览后购买是一个非常稀疏的图，节点的平均度数为 5。直接用 CNE 来建模如此稀疏的图是不明智的，因为可以用来训练的边太少，模型不能够得到充分地训练。针对这个任务，使用一种多任务学习的方式，构建两组编码器和损失函数，同时建模共同浏览和浏览后购买两种关系。可以观察到的是，在大多数情形下，对某个节点来说，属于浏览后购买关系的邻居节点是共同浏览关系的子集。可以利用共同浏览关系的编码器和损失函数来指导浏览后购买关系的建模。

如表5.4所示, 多任务模型 CNE$_{MUL}$ 与所有基线算法相比都有更加突出的表现，所有的基线算法都是在浏览后购买一种关系上训练的。和 CNE 相比，DeepWalk 也能够给出有竞争力的结果。其他的对比算法都处在欠拟合的状态。DeepWalk 由于模型的参数较少，因此图的稀疏性对其影响不是很明显。CNE 同 DeepWalk 有着类似的优势，由于编码器和特征表示向量是在所有节点上共享的，CNE 所涉及的模型参数也比较少。GraphSAGE 非常不适合节点度数很少的情形，原因在于 GraphSAGE 算法的核心在于融合邻域节点信息，而图稀疏，邻域节点过少时，GraphSAGE 算法也巧妇难为无米之炊。

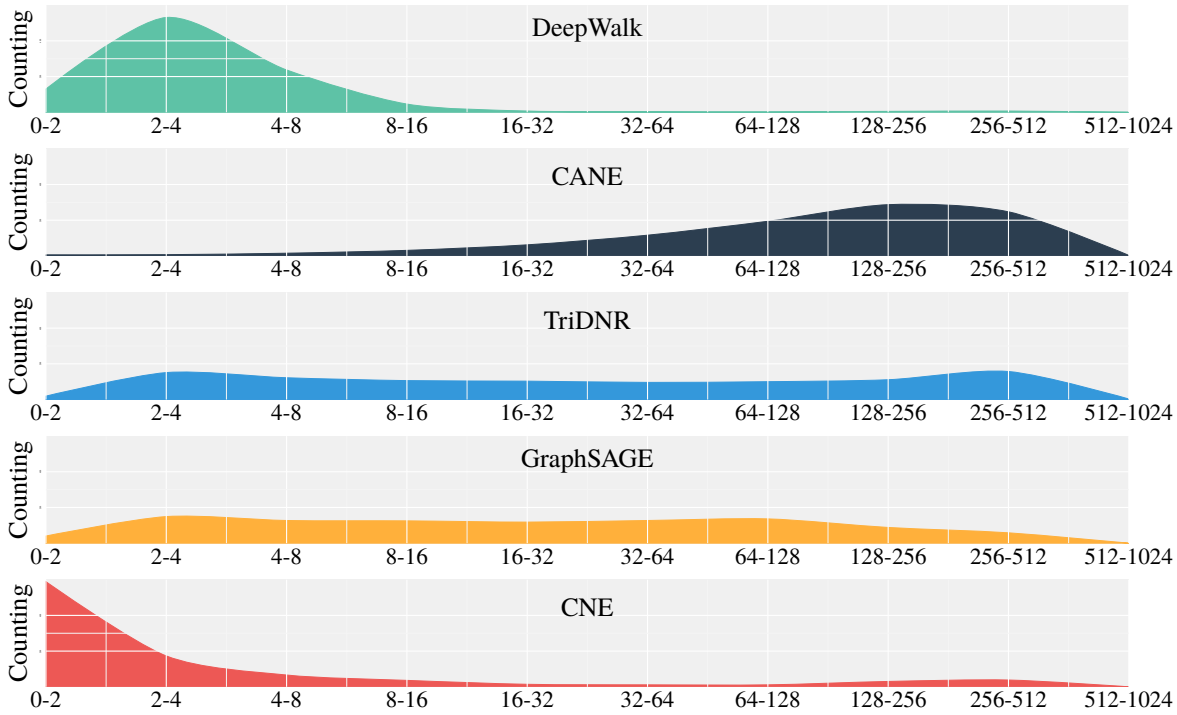


图 5.2　下一个点击的商品在相关性排序序列中位置的分布。

5.1.4.6　任务四：点异构图的边预测

本任务的设置为了证明 CNE 可以适用于多节点类型的图。实验前搜集了淘宝网上的用户行为序列，并构造了一个图。任务的目标是根据用户的前 n 个行为预测第 $n+1$ 个行为。本实验为基线算法构造了一个仅包含商品的同构图，如果两个商品被用户接连点击过，则在商品之间添加一条边。整个图包含 8744 个节点和 29976 条边。所有的基线算法会为每个商品学习得到一个向量表示，基于此计算得到一个用户历史行为的表示向量：对历史行为中包含的商品向量表示求和。同时实验也建立一个包含用户和商品的异构图，每个用户节点与其最近浏览过的 n 个商品相连。CNE 使用 n 个 GRU 编码器来处理 n 个商品，并将 n 个 GRU 的最后一个状态相加得到用户历史行为节点的表示向量。在实验中，$n = 4$。

与任务一到任务三相比，任务四更加具有挑战性。原因在于任务四中，每个预测任务只有一个正确答案。计算所有数据集中所有商品向量和历史行为向量的 cosine 相似度，并将相似度从大到小排序。在计算的过程中，出现在历史行为里的商品已经被过滤掉。测试了 1000 个用户的历史行为，统计正确答案在 1000 个序列中出现的位置。

图5.2展示正确答案在排序后的序列中的位置的分布 CNE 会把正确答案排在前三位，DeepWalk 会把正确答案排在稍微靠后的地方。同时还可以观察到，CNE 会把一部分正确答案排在相当靠后的地方。除此以外，其他基于节点属性的对比算法也会有类

表 5.5　用户的点击序列以及各个算法给出的排在前列的相关商品 [1]。

	Rank	Product Title
Click Record	1	Spring green loose mid-sleeve casual T-shirt.
	2	Pierced lace off shoulder 3/4 sleeve loose blouse.
	3	Plus size floral printed slimming princess dresses.
	4	Fake-two-piece pierced lace flowy tank blouse.
DeepWalk	1	Cotton plain loose white t-shirt.
	2	Spring and summer outlet high-waist shorts.
	3	Ethnic style Thailand Napal summer holiday long dress.
CANE	1	Original design fashion loose hip pants.
	2	Ethnic style Thailand Napal summer holiday long dress.
	3	Summer sleeveless wrinkled dress.
TriDNR	1	Puff sleeve elegant floral printed blouse.
	2	Extra size slimming pierced long scarf wrap shawl.
	3	Spring and summer sleeveless casual jumpsuits.
GraphSAGE	1	Korean summer beautiful dress.
	2	Hong-kong embroidery dress.
	3	Korean summer fashion v-neck hoodie.
CNE	1	Summer flower figure-flattering princess dress.
	2	Slimming cold shoulder empire waist fairy dress.
	3	Pink colorful dotted silk long-sleeve blouse.

[1] 相似词义的词语背景色是一致的。(large size , casual , feminine , hot weather)

似的现象。原因在于很多商品都有非常相似的属性，这使得在计算 cosine 相似度时，与历史行为中的属性相似的商品会被排在最前面。GraphSAGE，CANE 和 TriDNR 都在这个任务上表现很差，原因就在于商品相似的文本信息使他们混淆了。

表5.5展示了不同图表示学习算法的预测效果。表格的前四行是商品的名称以及点击的顺序。表示学习得到商品的向量表示和历史记录的向量表示。通过计算欧式距离对所有的商品做了排序，依次为接下来最有可能点击的商品。同时还做了过滤的操作，那些已经被点击过的商品不会出现在推荐点击的列表中。从用户的点击记录来看，该用户正在搜索大码 (*loose*, *plus size*)，休闲 (*casual*, *floral printed*, *flowy*)，女性化的 (*off shoulder*, *princess*, *pierced lace*)，炎热天气的 (*T-shirt*, *blouse*, *dress*, *3/4 sleeve*) 衣服。对比算法给出的推荐商品只涵盖了部分的用户目标。GraphSAGE 和 DeepWalk 无法将相似的商品排在前面。CANE 和 TriDNR 把有相同词汇的商品排在了前面。CNE 则可以找到语义相似的商品，并且包含了用户在点击序列中展现的全部需求：大码 (*figure-flattering*, *slimming*), 休闲的 (flower, *colorful*, *dotted*), 女性化的 (*princess*, *fairy*, *cold shoulder*, *pink*), 炎热天气的 (*dress*, *silk*). 这个例子说明 CNE 可以很好的捕捉商品之间的语义相似性，并

识别用户的意图。

5.2 一个基于随机游走的节点中心度指标

5.2.1 应用背景

节点的中心度是网络科学领域中一个古老而重要的概念。基于节点中心度指标挖掘出的中心节点在迅速传播信息，有效防治传染病，深入理解图中的层级结构等问题中起到关键作用[133,161]。根据节点中心度在计算时所涉及的节点范围，节点中心度可以被分为两类：局部中心度和全局中心度。以节点度数和聚合系数为代表的局部中心度简单却有效，其计算只依赖目标节点的自我网络。与之相反，当要考虑和信息传播、影响力最大化等相关的任务时，则需要在更大的范围内考察节点的传播能力。中介数 Betweenness 和接近中心性 Closeness 都是从图全局的角度来刻画节点中心度。由于这些指标是在全图结构的基础之上进行计算的，他们会富含更丰富的信息，也因此在图分析任务中被广泛使用[161]。然而，对于大规模的图来说，精确计算这些全局中心度往往是不可能完成的，全局中心度的近似算法在实际应用中的表现也不尽如人意。同时，全局中心度的定义也与实际应用之间存在一定偏差，例如中介数的定义基于图上的最短路径，然而图中的很演化或者传播活动是不存在任何目的的[80]，相较最短路径等这些理想最优路径，随机游走更符合现实。这也是基于随机游走的中心度在很多任务上有更好表现的理由之一[162]。

在本节中提出了一个新的中心度指标，Node Conductance，衡量的是节点在随机游走中被再次访问到的可能性。根据定义，Node Conductance 是从目标节点的角度来衡量这个图的连接状况。同时，正如上文所讲，Node Conductance 没有限定最优路径，而是基于随机游走路径，因此更加适合实际任务。从直观的角度上讲，Node Conductance 综合了节点度数和中介数。度数高的节点有更高的概率在短随机游走路径中被重复访问；中介数高的节点在较长的随机游走路径中更容易被访问。本节进一步证明 Node Conductance 可以通过目标节点和它多阶邻域共同组成的导出子图近似得到。换句话说，Node Conductance 可以利用比较短的随机游走路径得到一个近似的中心度。这一发现可以快速高效地完成大图上的 Node Conductance 近似值。

本节进一步关注 Node Conductance 的近似算法，也就是说节点在短随机游走路径中被重复访问的可能性。更确切地说，本节扩展了基于 word2vec 的图表示学习模型的理论基础，发现了节点表示向量，图结构，和近似 Node Conductance 值之间的关系。图表示学习的目标是刻画图中节点之间的复杂关系。在众多的算法之中，DeepWalk 是第一个将 word2vec 模型引入图表示学习任务中的算法。本节认为 DeepWalk 的工作是统计节点对在短随机游走路径中共同出现的概率，这一点与 Node Conductance 的定义

不谋而合。由于在随机游走中，节点每一个决策都与当前位置的节点度数相关，所以一对节点在随机游走共同出现的概率一定是与他们之间的局部拓扑结构相关的。Node Conductance 关注的是一对特殊节点共同出现的概率 — 目标节点和自身共同出现的概率。Node Conductance 和 DeepWalk 有十分相似的直观意义，因此利用 DeepWalk 的结果来计算 Node Conductance 的近似值是可行的。

5.2.2　指标的形式化表示

Conductance 也是一种刻画图的连边特点的度量，它计算的是在随机游走中，离开一群节点的难度[163]。此处将提出的新指标命名为 Node Conductance，用来表示随机游走中，离开某一个节点的难度。

考虑一个无向图 G，方便起见，认为 G 是无权图，尽管以下的所有结论都可以推广到带权图上去。G 上的随机游走是一个马尔可夫过程。将 Node Conductance 定义为节点 i 在随机游走中被重复访问的概率，这一概率记做 $P_r(i|i)$。在接下来的章节中，将证明两个节点在随机游走的窗口中共同出现的概率由两个节点共同属于的子网所决定。Node Conductance 考虑的是某个节点和它自身共同出现的概率，因此 Node Conductance 可以衡量一个节点周围的连接情况是否紧密。

考虑一个无向图 G，图中包含 n 个节点和 m 条边。邻接矩阵 $\boldsymbol{A}$ 是对称的，其中值为 1 的元素代表两点之间有边相连。向量 $\boldsymbol{d} = \boldsymbol{A}\mathbf{1}$，其中 $\mathbf{1}$ 是一个 $n \times 1$ 的向量，每个位置的值代表节点的度数。图 G 中的随机游走在每个节点都以相等的概率跳转到邻居节点。

$$\boldsymbol{p}_{t+1} = \boldsymbol{p}_t \boldsymbol{D}^{-1}\boldsymbol{A} \equiv \boldsymbol{p}_t \boldsymbol{M}, \tag{5.5}$$

其中 $\boldsymbol{p}$ 是概率向量 $\boldsymbol{D}$ 是以度数为对角线的对角矩阵 $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{d})$。$\boldsymbol{M}$ 是转移矩阵。令 $\pi = \pi\boldsymbol{M}$，随机游走的稳定状态是 $\pi = \boldsymbol{d}^{\top}/2m$。

对于一个起始于 i 节点的随机游走，在第 r 步发现游走至 j 节点的概率是：

$$P(j, r|i) = [\boldsymbol{M}^r]_{ij} \tag{5.6}$$

Node Conductance $P_r(i|i)$ 计算的是节点 i 在 r 步之内被访问的概率 (步长小于 r 步的情形也被考虑)。

$$P_r(i|i) \propto \sum_{s=0}^{r} P(i, s|i) \propto \boldsymbol{P}_{ii}^{(r)}, \quad \boldsymbol{P}^{(r)} = \sum_{s=0}^{r} \boldsymbol{M}^s. \tag{5.7}$$

假设 r 接近于无限，$P_\infty(i|i)$ 此时成为了一个全局的中心性度量，

$$\begin{aligned}\boldsymbol{P}^{(\infty)} &= \sum_{s=0}^{\infty} \boldsymbol{M}^s = (\boldsymbol{I} + \boldsymbol{M} + \boldsymbol{M}^2 + \cdots + \boldsymbol{M}^\infty) \\ &= (\boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A})^{-1} = (\boldsymbol{D} - \boldsymbol{A})^{-1}\boldsymbol{D}.\end{aligned} \tag{5.8}$$

$\boldsymbol{D}-\boldsymbol{A}$，即 *Laplacian* 矩阵 $\boldsymbol{L}$ 是非平凡的，不可以直接求逆。引入伪逆的概念：

$$\boldsymbol{L}_{ij} = \sum_{k=1}^{N} \lambda_k u_{ik} u_{jk}, \tag{5.9}$$

其中 $\boldsymbol{\lambda}$ 和 $\boldsymbol{u}$ 分别是特征值和特征向量。向量 $[1, 1, ...]$ 一定是特征值 0 对应的特征向量，伪逆 $\boldsymbol{L}^\dagger$ 的特征值被定义为：

$$g(\lambda_k) = \begin{cases} \frac{1}{\lambda_k}, & \text{if } \lambda_k \neq 0; \\ 0, & \text{if } \lambda_k = 0. \end{cases} \tag{5.10}$$

$P_\infty(i|i)$ 只考虑对角线上的元素，

$$\boldsymbol{L}^\dagger_{ii} = \sum_{k=1}^{N-1} \frac{1}{\lambda_k} u_{ik}^2, \quad P_\infty(i|i) \propto \boldsymbol{L}^\dagger_{ii} \cdot d_i, \tag{5.11}$$

其中 d_i 是节点 i 的度数，是 $\boldsymbol{d}$ 中的第 i 个元素的值。

Node Conductance 的窗口大小可以被设定为任意值，小窗口对应局部的节点中心性，大窗口对应全局的节点中心性。然而实际上，无论窗口值设为多少，Node Conductance 的值还是与局部的拓扑结构更加相关。正如 Eq.5.7展示的那样，$\boldsymbol{M}^s$ 中元素的值随着 s 的增大会迅速变小。随着随机游走步长的增长，重新回到起始点的概率会越来越小。因此，没有必要为 Node Conductance 设置一个过大的窗口大小。Node Conductance 的形式化表示仍需要矩阵的幂乘和求逆。为了适用于大规模的图，Node Conductance 仍需要进一步的近似计算，以降低计算复杂度。接下来将证明 Node Conductance 可以被图的局部很好的近似。

5.2.3 新指标、子图中心度与 PageRank

从定义上看，Node Conductance 与子图中心度 Subgraph Centrality（SC）和 PageRank（PR）十分接近。实际上，Node Conductance 只考虑从目标节点出发并结束的游走路径。PageRank 计算的是随机游走的稳定分布，它考虑的是从任意节点出发的，无限步数的，到达目标节点的随机游走路径：

$$\text{PR} = \boldsymbol{D}(\boldsymbol{D} - \alpha\boldsymbol{A})^{-1}\boldsymbol{1}, \tag{5.12}$$

其中 α 是随机跳到图中任意一个节点的概率。式子5.12和式子5.8之间的区别就在于所考虑的随机游走不同。在式子5.12中，通过与矩阵 $\mathbf{1}$ 相乘，i 节点的 PR 值等于 $\boldsymbol{D}(\boldsymbol{D}-\alpha\boldsymbol{A})^{-1}$ 第 i 行元素的和。在式子5.8中，i 节点的 NC 值是矩阵的第 i 行 i 列位置的元素。总的来说，Node Conductance 更关注节点的邻域特征，而 PageRank 更关注全局的图结构。这一区别使得 PageRank 在信息检索领域更为有效，但在社会网络分析中却很难发挥优势。毕竟，社会行为在大多数情况下与全局的影响力是无关的，而与局部的关系更强。

SC 计算的是目标节点参与的子图个数，这个数字与从目标节点出发并结束的封闭路径数目是相同的。为了使得 SC 指标的计算是收敛的，其作者又引入了一个因子（式5.13中的分母），然而这使得新的 SC 值很难被解释。

$$\mathrm{SC}(i)=\sum_{s=1}^{\infty}\frac{(\boldsymbol{A}^s)_{ii}}{s!}. \tag{5.13}$$

与之相反，NC 的概念很容易理解，而且它从定义上讲是收敛的。

5.2.4　利用局部子图近似计算中心度

上文中只是简单说明 Node Conductance 的值与节点所处的局部子图更为相关。接下来将证明 Node Conductance 可以被局部子图很好的近似。

定义 6 (局部子图) *对于图 $G=(V,E)$ 来说，给定节点 $x\in V$, 节点 x 的 h-跳局部子图指的是由节点集合 $\{i|d(i,x)\le h\}$ 组成的导出子图 G_x^h，其中 $d(i,x)$ 是节点 i 和 x 之间的最短路径长度.*

定义 7 (γ-衰减指标) *对于节点 x 来说，γ-衰减指标有如下特征：*

$$\mathcal{M}(x)=\sum_{l=1}^{\infty}\gamma^l f(x,l), \tag{5.14}$$

其中 γ 是一个 0 到 1 之间的衰减因子，f 是一个关于 x 和 l 的非负函数。

在某些情况下，一个 γ-衰减指标是可以通过 h-跳局部子图近似计算得到的，近似误差随着 h 的增长至少指数型下降。

定理 4 *给定一个 γ-衰减指标 $\mathcal{M}(x)=\sum_{l=1}^{\infty}\gamma^l f(x,l)$，如果 $f(x,l)$ 符合一下两个要求:*

- *$f(x,l)\le\lambda^l$，其中 $\lambda<1/\gamma$,*
- *$f(x,l)$ 可以利用 G_x^l 被计算出来，*

那么 $\mathcal{M}(x)$ 可以通过 G_x^h 近似计算得到，且近似误差随着 h 的增长至少指数型下降。

证明 4 可以通过对一个 γ-衰减指标的前 h 项求和，以实现对该指标的近似计算。

$$\widetilde{\mathcal{M}}(x) = \sum_{l=1}^{h} \gamma^l f(x,l). \tag{5.15}$$

近似误差的边界通过计算两者的差得到:

$$|\mathcal{M}(x) - \widetilde{\mathcal{M}}(x)| = \sum_{l=h+1}^{\infty} \gamma^l f(x,l) \leq \sum_{l=h+1}^{\infty} \gamma^l \lambda^l = \frac{(\gamma\lambda)^h}{1-\gamma\lambda} \tag{5.16}$$

在具体的应用场景下，一个小的 $\gamma\lambda$ 值会使得近似误差快速下降。

节点 x 的 Node Conductance 值被定义为该点在第 s 步被随机游走重复访问的概率之和，s 的取值接近无限：

$$\mathtt{NC}_{\infty}(x) = \sum_{s=1}^{\infty} P(x|x,s) = \sum_{s=1}^{\infty} (\boldsymbol{D}^{-1}\boldsymbol{A})^s_{xx}. \tag{5.17}$$

如果将 Node Conductance 整理为 γ-衰减指标的形式，那么 $\gamma = 1$ 且 $f(x,l) = (\boldsymbol{D}^{-1}\boldsymbol{A})^l_{xx}$.

定理 5 对于任意节点 x，$(\boldsymbol{D}^{-1}\boldsymbol{A})^l_{xx}$ 是一个符合定理4的 γ-衰减指标。

证明 5 $(\boldsymbol{D}^{-1}\boldsymbol{A})^l_{ij}$ 是随机游走以 i 为起点，l 步后达到 j 点的概率。

$$\sum_{j\in V} (\boldsymbol{D}^{-1}\boldsymbol{A})_{ij} = 1. \tag{5.18}$$

因此，$(\boldsymbol{D}^{-1}\boldsymbol{A})^l_{xx} \leq 1$，等号仅在图由两个连接的节点组成且 l 为偶数时成立。除此之外，$(\boldsymbol{D}^{-1}\boldsymbol{A})^l_{xx} < 1$ 且 $f(x,l) < 1/\gamma$ 是成立的。

5.2.5 新指标与图表示学习的关系

SGNS 的损失函数如下[164,165]：$\mathcal{V}_W$ 是词表，i 是目标词，$\mathcal{V}_C$ 是它的上下文，$\#(i,j)$ 是 j 出现在以 i 为中心的窗口内部的次数。$\#(i)$ 是 i 在文本中出现的次数。$\boldsymbol{w}_i$ 和 $\boldsymbol{c}_i$ 是 i 的输入和输出向量。

$$\begin{aligned} l = &\sum_{i\in\mathcal{V}_W}\sum_{j\in\mathcal{V}_C} \#(i,j)\big(\log\sigma(\boldsymbol{w}_i\cdot\boldsymbol{c}_j)\big) \\ &+ \sum_{i\in\mathcal{V}_W} \#(i)\Big(k\cdot\sum_{\mathrm{neg}\in\mathcal{V}_C} P(\mathrm{neg})\log\sigma(-\boldsymbol{w}_i\cdot\boldsymbol{c}_{\mathrm{neg}})\Big). \end{aligned} \tag{5.19}$$

neg 是根据 $P(i) = \#(i)/|D|$ 采样得到的负例，D 是所有观测到的词。SGNS 的目标是去优化损失函数 l。定义 $x = \boldsymbol{w}_i\cdot\boldsymbol{c}_j$，并计算它对于 l 的偏导数：

$$\frac{\partial l}{\partial x} = \#(i,j)\cdot\sigma(-x) - k\cdot\#(i)\cdot P(j)\sigma(x). \tag{5.20}$$

令导数等于 0，得到：

$$\boldsymbol{w}_i\cdot\boldsymbol{c}_j = \log\Big(\frac{\#(i,j)}{\#(i)\cdot P(j)}\Big) - \log k, \tag{5.21}$$

表 5.6　Node Conductance 和已有中心度的排序的相关性。

指标	COEF	Karate	word	football	jazz	celegans	email	polblog	pgp
Degree	ρ	0.95	0.98	0.51	0.98	0.91	0.99	0.99	0.95
	R^2	0.87	0.96	0.41	0.95	0.66	0.78	0.92	0.01
NC_∞	ρ	0.93	0.98	0.41	0.98	0.89	0.99	0.99	0.95
	R^2	0.87	0.96	0.31	0.95	0.66	0.78	0.05	0.01
Subgraph Centrality	ρ	0.71	0.91	0.48	0.85	0.66	0.87	0.95	0.31
	R^2	0.83	0.74	0.37	0.67	0.43	0.34	0.53	0
Closeness Centrality	ρ	0.79	0.87	−0.10	0.84	0.45	0.88	0.92	0.32
	R^2	0.54	0.68	0.03	0.68	0.27	0.46	0.42	0
Network Flow Betweenness	ρ	0.91	0.94	0.01	0.82	0.81	0.96	–	0.91
	R^2	0.83	0.94	0.01	0.57	0.56	0.78	–	0.01
Betweenness	ρ	0.84	0.89	−0.04	0.70	0.77	0.89	0.89	0.81
	R^2	0.62	0.76	0.02	0.27	0.27	0.81	0.66	0.01
Eigenvector Centrality	ρ	0.64	0.90	−0.33	0.85	0.66	0.87	0.95	0.30
	R^2	0.76	0.86	0.18	0.75	0.66	0.53	0.66	0
PageRank	ρ	0.96	0.98	0.48	0.97	0.83	0.97	0.97	0.92
	R^2	0.86	0.96	0.40	0.90	0.62	0.79	0.95	0.03
Clustering Coefficient	ρ	−0.45	0.37	0.22	−0.33	−0.65	0.33	0.20	0.59
	R^2	0.22	0.001	0.18	0.07	0.27	0.01	0.006	0.0001

其中 k 是负采样的个数。

对于某个节点 i，计算它的输入向量和输出向量的乘积:

$$\boldsymbol{w}_i \cdot \boldsymbol{c}_i = \log(\frac{\#(i,i)}{\#(i) \cdot P(i)}) - \log k. \tag{5.22}$$

通常这一值会用实际观察到的频次来估计。

$$\begin{aligned} \boldsymbol{w}_i \cdot \boldsymbol{c}_i &= \log(\frac{\#(i,i)}{\#(i) \cdot P(i)}) - \log k = \log(\frac{P_r(i,i)}{P(i) \cdot P(i)}) - \log k \\ &= \log(P_r(i|i) \cdot \frac{1}{P(i)}) - \log k. \end{aligned} \tag{5.23}$$

$P_r(i,i)$ 是节点 i 在窗中出现两次的概率。$P(i)$ 是节点在随机游走中被访问到的概率，这一概率与节点的度数正相关。

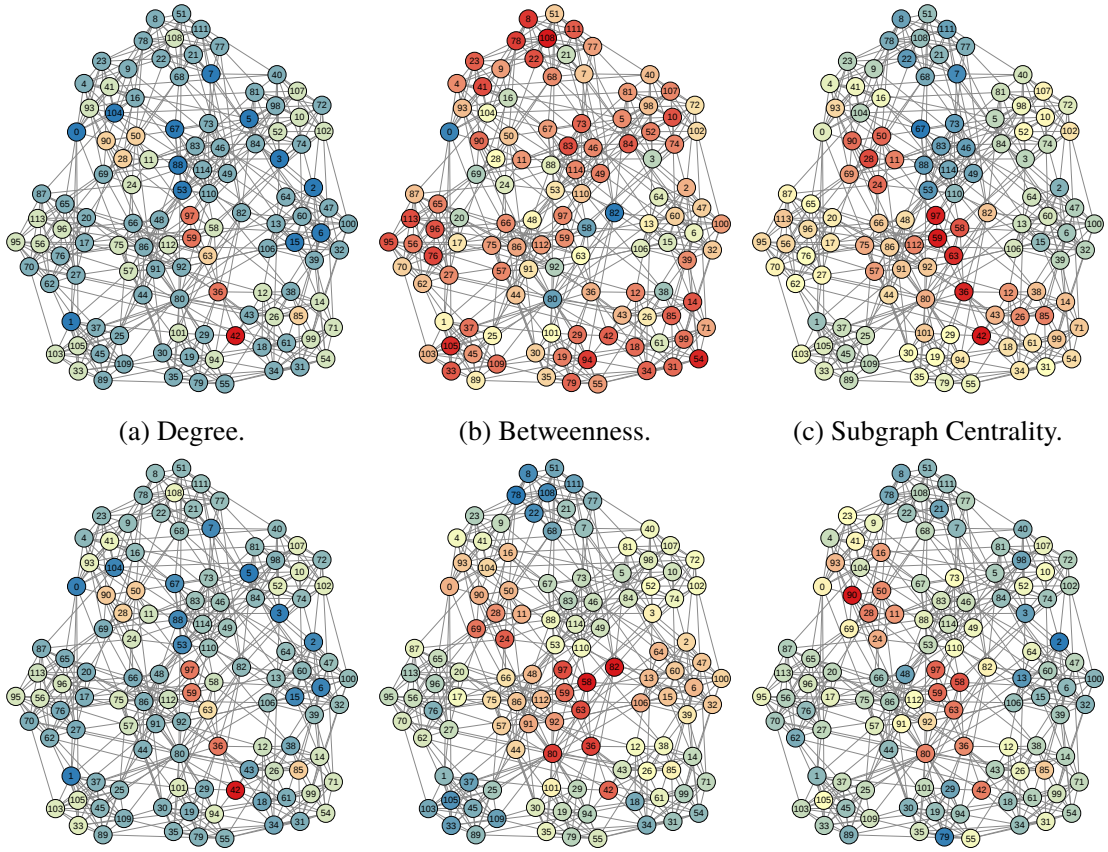
$$P_r(i|i) = \exp(\boldsymbol{w}_i \cdot \boldsymbol{c}_i) \cdot k \cdot P(i) \propto \exp(\boldsymbol{w}_i \cdot \boldsymbol{c}_i) \cdot \deg(i). \tag{5.24}$$

在接下来的实验中，$\exp(\boldsymbol{w}_i \cdot \boldsymbol{c}_i) \cdot \deg(i)$ 的值即作为节点 i 的相对 Node Conductance 值。在一系列的排序任务中，不需要 Node Conductance 的精切值，有他们的相对关系即可。

5.2.6 新指标与已有中心度的对比

虽然不同的中心度的出发点不同，刻画的节点属性不同，已经有实验证明他们之间存在一定的相关性[166]。在多个数据集上计算了不同中心度的相关性。Node Conductance (window = ∞) 是有公式5.11计算得到的。Node Conductance (window = 6) 是由窗口大小为 6 的 DeepWalk 算法得到的。如在表5.6中展示的那样，从两个角度计算了不同指标的相关性：Spearman 系数和 R 方系数。前者只考虑节点排序的相关性，而后者还考虑了中心性的值的相关性。除了 football 数据集，Degree, Node Conductance (window = ∞) 和 PageRank 都与 Node Conductance (window=6) 有较强的相关性。

将特例 football 绘制出来，已对不同的中心度度量有更直观的认识。同时也行进一步探究为什么 Node Conductance 在这个数据集上与其他指标的相关性极弱。图5.3 展示了 football 图。节点的颜色代表节点中心度排序。中心度低的节点是红色，中心度中的黄色，中心度高的是蓝色。



(a) Degree. (b) Betweenness. (c) Subgraph Centrality.

(d) PageRank. (e) NC (window = ∞). (f) NC (window = 6).

图 5.3 被不同节点中心度标记的足球图数据集。

对比图5.3a 图5.3b 和图5.3f，可以发现 Node Conductance (window = 6) 的结果似乎合并了 Degree 和 Betweenness 的结果。Node Conductance 给低 Degree 和高 Betweenness

表 5.7　静态图数据集的基本信息。

数据集	节点	边	社区数目	聚合系数
DBLP	317K	1M	13K	0.63
Amazon	335K	926K	75K	0.40
Youtube	1.1M	3.0M	8K	0.08

表 5.8　Flickr 数据集的四个快照。

快照	节点	边	快照	节点	边
1	1,487,058	11,800,425	2	1,493,635	11,860,309
3	1,766,734	15,560,731	4	1,788,293	15,659,308

的节点一个较低的值。这直观的展示了 Node Conductance 可以捕捉到局部和全局的结构信息。PageRank (图5.3d) and Degree (图5.3a) 也展现了极强的相关性。

在节点簇中的节点有着较低的 Subgraph Centrality (图5.3c)，但较高的 Node Conductance (图5.3f)，尤其当节点簇比较孤立，鲜有边与其他点相连。Subgraph Centrality 计算的是走回目标节点的路径数目，而 Node Conductance 计算的是重返的概率。两者之间存在着负相关的关系。

当窗口大小增大时，节点颜色的分布基本一致（图5.3e 和图5.3f）。图5.3e中，某些节点簇的值变小了。这是由于窗口所圈定的粒度不同导致的。

5.2.7　实验评测

在接下来的实验中，使用 DeepWalk 计算得到的 Node Conductance，以验证其在静态图和动态图上的重要作用和计算效率。实验测试了多个窗口大小，最终发现窗口大小为 6 时效果最优。实验尽可能精确地计算每一个对比节点中心度，但是其中的有些仍然无法在大数据集上实现。

5.2.7.1　实验准备

- **带有节点社区属性的静态图**在实验中选取了学术合作图 DBLP，电商共同购买图 Amazon，线上社区网络 Youtube 三个图数据集[128]。在 DBLP 中，两个作者如果互为共同作者，那么就创建一条边。发表论文的会议被视为节点的社区。DBLP 网络十分稠密，因此它的聚合系数在三个数据集中最高。在 Amazon 网络中，边的含义是两个商品经常被用户同时购买。社区指的是商品属于同一类目。Youtube 中用户可以自主选择加入不同的兴趣小组，兴趣小组在数据集中被视为社区，用户之间的关注关系是网络中的连边。Youtube 网络十分稀疏，它的聚集系数是最小的。

表 5.9　全局节点中心度的运行时间（秒）。

数据集	AP[1]	NC[2]	AB[3]	AE[4]	SC[5]	FB[6]
DBLP	914	985	14268	–	–	–
Amazon	941	988	9504	–	–	–
Youtube	2883	3464	168737	–	–	–

[1] approximate PageRank.　[2] Node Conductance.
[3] approximate Betweenness.　[4] approximate Eigenvector Centrality.
[5] Subgraph Centrality.　[6] Network Flow Betweenness.

表 5.10　各个节点中心度的 Spearman 排序相关系数 ρ[7]。

数据集 Datasets	ρ_{NC}	ρ_{D}	ρ_{AB}	ρ_{AE}	ρ_{AP}	ρ_{CC}
DBLP	**0.62**	0.60	0.61	0.59	0.48	-0.29
Amazon	**0.28**	0.27	0.17	0.15	0.23	0.007
Youtube	**0.26**	0.24	0.23	0.21	0.20	0.22

[7] ρ 的角标代表不同的中心度。D: 度数. 其他的角标与上表保持一致。

- **动态图** Flickr 图[167]是在 2006 年 11 月 2 日到 2007 年 5 月 18 日之间被收集，对应了其 104 天的规模增长。如表5.8所示，在这段时间中有 4 个图快照被保存下来。整个无向无权图包含大概 300,000 新用户和超过 380 万的新增边。也就说，相对于第一个图快照，整个图增长了 20% 的用户和 32% 的边。

5.2.7.2　运行效率

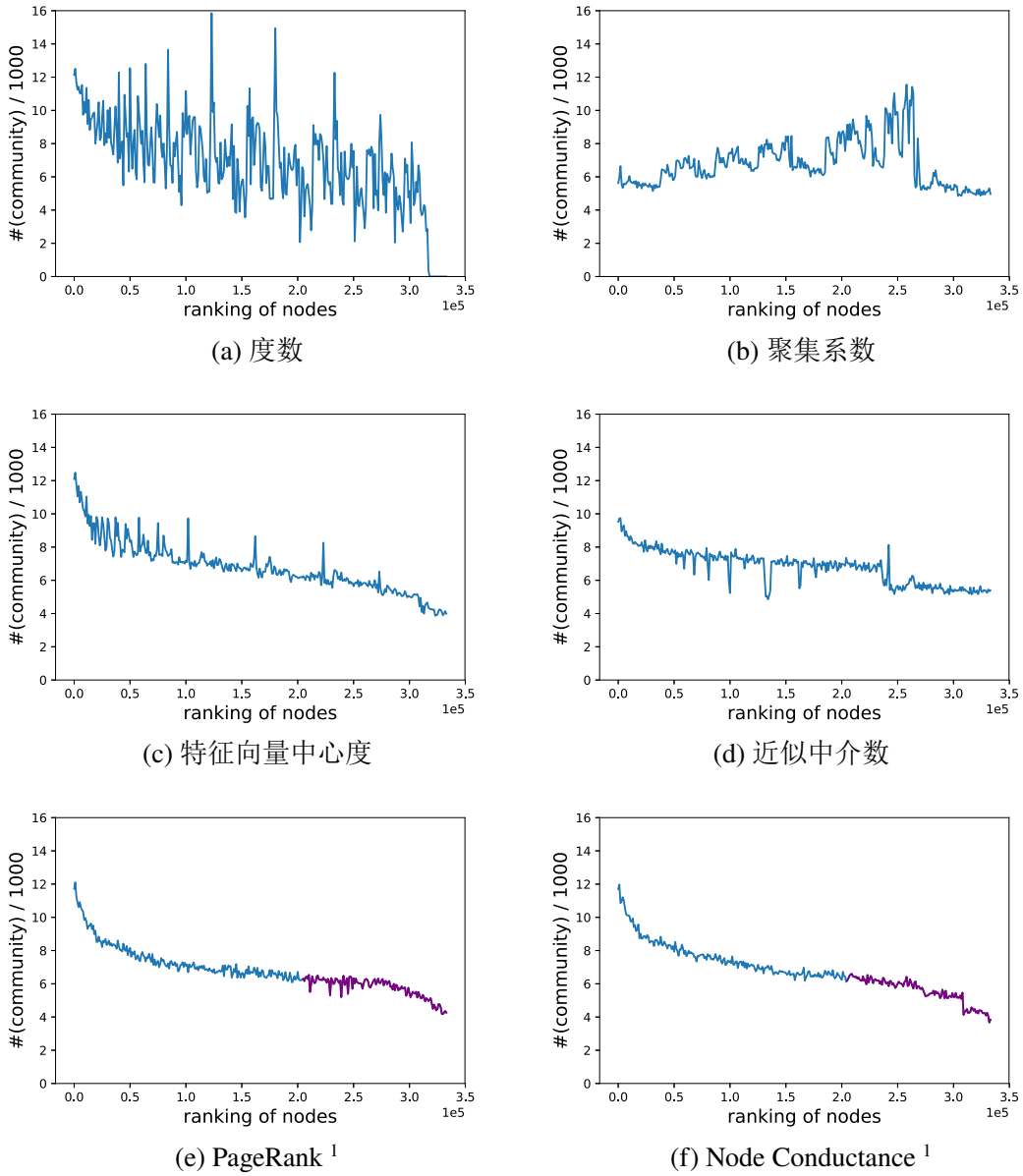
实验所使用的计算机配置是：两个 Intel(R) Xeon(R) CPU E5-2620 at 2.00GHz, 64GB RAM。Node Conductance 由 DeepWalk 计算得到，其参数设置为 m=80, l=40, w=6,and d=128, 与 DeepWalk 论文[10]中的设置保持一致。由于 Node Conductance 实际是 DeepWalk 算法的副产品，因此 Node Conductance 的运行时间与 DeepWalk 是一致的。实验中，特征向量中心度和 PageRank 是通过迭代法近似计算得到的，实验将迭代误差阈值设定为 $1e{-}10$。中介数的近似计算通过随机选择 1000 个锚节点来实现，选择更多的锚节点会消耗更多的时间。由研究[168]表明，在近似计算中介数时，在整个图中随机选择锚节点可以取得最好的效果。子图中心度和网络流中介数目前还没有被广为接受的近似算法。

表 5.9中展示的全局中心度的运行时间。近似特征值中心度，子图中心度和网络流中介数在三个数据集上都不能在合理的等待时间内完成计算。由 DeepWalk 计算得到的 Node Conductance 与近似 PageRank 有着相似的计算效率，同时两者远远高于近似中介数的运行效率。与当下被广泛使用的全局中心度相比，通过 Deepwalk 计算得到的 Node Conductance 可扩展性更强，在大规模图上的能力更加强大。

5.2.7.3　寻找跨多个社区的节点

本节利用 Node Conductance 来识别横跨多个社区的节点。有时，这种节点也被叫做跨结构洞节点。Amazon, DBLP 和 Youtube 数据集都提供了节点的社区属性信息，因此可以统计每个节点参与的社区数目。在实验中，图中节点根据所计算出的节点中心度从大到小排列。

实验中首先计算了两个排序后的序列之间的相似度，根据节点中心度值的排序序



(a) 度数　(b) 聚集系数

(c) 特征向量中心度　(d) 近似中介数

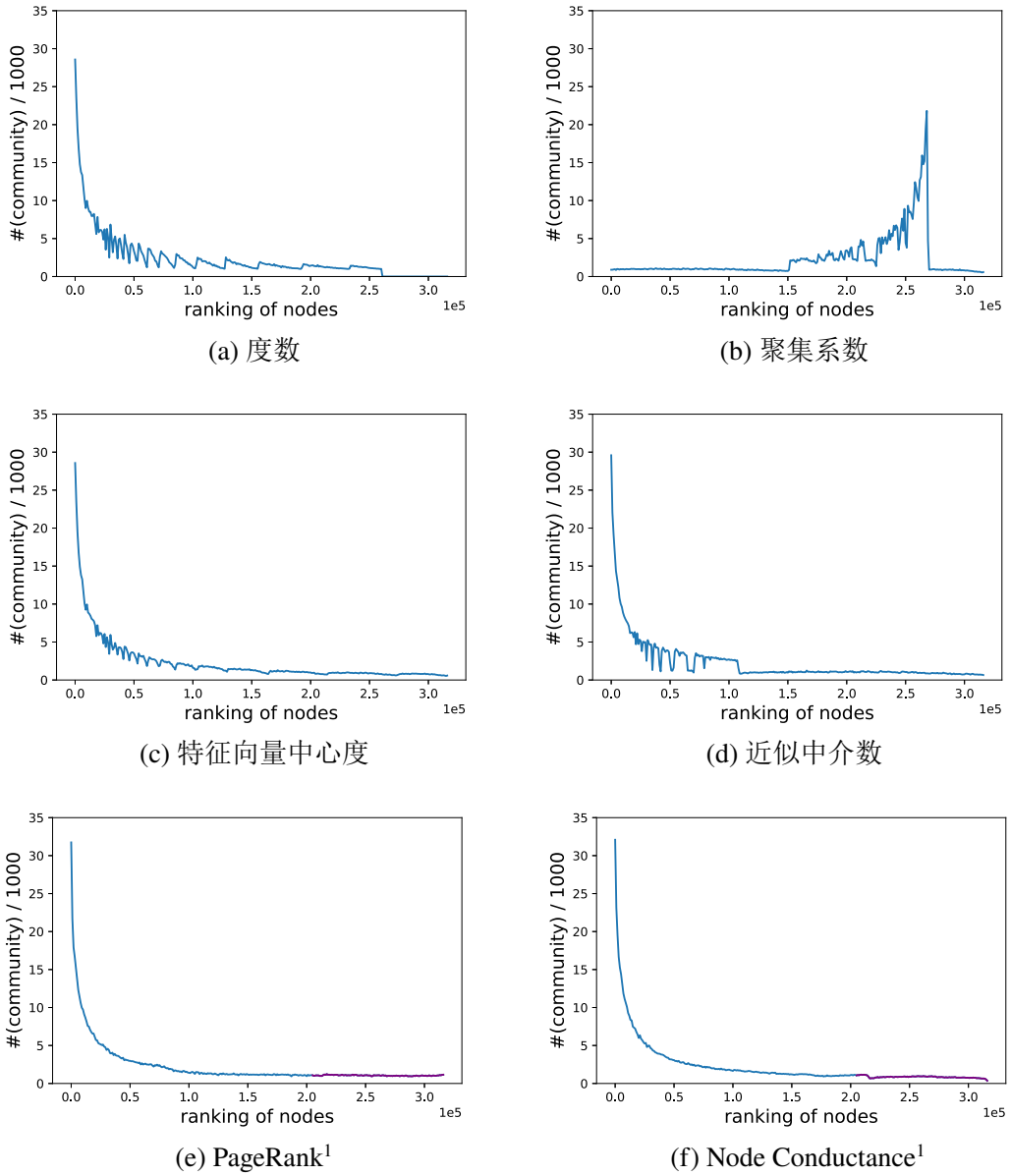
(e) PageRank [1]　(f) Node Conductance [1]

[1] 最后两个图像的尾部被标记为紫色，以强调两条曲线之间的区别。

图 5.4　节点所属的社区数目 (Amazon 数据集) 与节点的不同中心度。

列和根据节点所属社区数目的排序序列。近似特征向量中心度在计算时，计算误差阈值设为 $1e{-}6$，其他参数设定与运行效率部分保持一致。相关性的结果展示在表5.10中，Node Conductance 的表现最优，PageRank 出乎意料的表现最差。

实验进一步探究了不同中心度给出的排序结果之间的差别，并将结果绘制出来：Y-轴是节点所属于社区的数目，X-轴是节点中心度的值。为了让曲线比较平滑，实验以 1000 为步长，计算每 1000 个节点所属社区数目的平均值。举例来说，曲线上坐标为 (x, y) 的点代表了中心度排在 $(1000x)$ 到 $(1000(x+1))$ 之间的节点平均属于 y 个社区。在图 5.4和图5.5中，所有的六个指标都可以在一定程度上反映出跨社区数目下降



(a) 度数　(b) 聚集系数

(c) 特征向量中心度　(d) 近似中介数

(e) PageRank[1]　(f) Node Conductance[1]

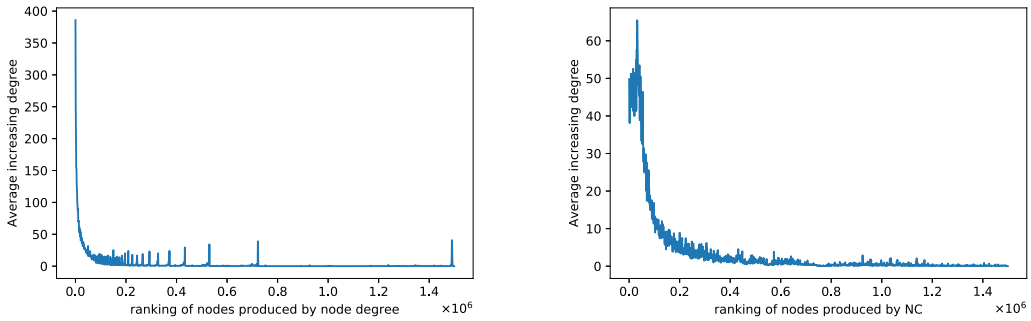
[1] 最后两个图像的尾部被标记为紫色，以强调两条曲线之间的区别。

图 5.5　节点所属的社区数目 (DBLP 数据集) 与节点的不同中心度。

的趋势。很显然，和其他中心度相比，Node Conductance 的曲线最为平滑，这表明了其在刻画节点地位上的独特优势。

讨论

- 节点度数和 PageRank 在表5.10中和图5.4&5.5中的表现看起来截然不同。在数据集中，很多节点所属于的社区数目是相同的，也就是说这些节点在社区数目排序的序列中序号应当是相同的。节点度数也有类似的情形，很多节点的度数是一致的，在度数排序序列中序号相同。然而在其他的节点中心度排序中，节点的中心度值都各不相同，也对应着不同的序号。由于这个原因，节点度数在计算排序的相关性时有更



(a) 节点度数　　(b) Node Conductance

图 5.6　新边生成的优先连接机制。

大的优势取得更高的相关性分数，但是在图5.4&5.5中的表现却又不尽如人意。对于 PageRank 曲线来说，它的尾部与 Node Conductance 的尾部有很大差别。在图5.4e中，PageRank 的尾部并不平滑，在图5.5e中，PageRank 的尾部还有一点小的上扬。这也就是说，PageRank 对于大部分的不太活跃的节点的鉴别能力较弱，因此在表5.10中，PageRank 的分数不高。

- Node Conductance 的计算完全基于图结构，然而节点的社区属性是完全取决于数据集的领域和应用背景的。节点的社区属性可以在一定程度上反应图的拓扑结构，而 Node Conductance 恰好有更强的能力捕捉到这一点。

5.2.7.4　新边生成的机制

这部分实验关注网络规模增长的机制。关于图扩张的现象，一个被广为接受的理论是“优先连接机制”[169]。它指的是图中旧节点与新节点建立边的概率与老节点的度数正相关。

实验关注 Flickr 图[167] 在 2006 年 12 月 3 日到 2007 年 2 月 3 日之间的图扩张。需要指出的是，接下来的结论在多个图快照上都成立，在论文中只展示了其中一例。实验将第一个快照中的图节点按照度数降序排列。同时也统计了在第二个快照中，每个节点新增的边数。图5.6a显示，节点度数和新增边数之间确实存在很强的优先连接性质。但是，整条曲线存在几处波峰，而且波动比较大（峰值是 50 左右），多次出现，不容忽视。图5.6b展示的是新增边数和节点 Node Conductance 之间的关系。比较两条曲线的左侧部分，Node Conductance 并不能成功识别那些有巨大度数变化的节点。但从另一方面看，Node Conductance 的曲线整体平滑很多，在尾部也没有出现明显的波动。这说明节点度数的优先连接机制更适用于那些度数较高的节点，而对于那些节点度数本身就比较小的节点来说，本实验说明基于 Node Conductance 的优先连接机制更符合这类节点的表现。

5.3 本章小结

本章重点关注路径解构的问题。首先提出基于路径解构的组合式图神经网络算法 — CNE。CNE 将“两个节点在图上连接的紧密度计算”分解为“判断两个节点在同一随机游走窗口内是否共同出现”的子问题。每个子问题只需关注一个相关的随机游走路径窗口。CNE 本身也是一个灵活度很高的算法框架。在实验中，算法在同构图，冷启动节点，边异构，节点异构四种类型的图上分别进行了验证，皆取得较好的效果。该工作已经发表在 RecSys2019 上。

随后，本文进一步深挖图表示学习中路径解构的数学含义，并基于此提出了一个适用于大规模图的全局性节点中心度指标 — Node Conductance。节点 x 的 Node Conductance 值被定义为该点在第 s 步被随机游走重复访问的概率之和，s 的取值接近无限。本文证明了该值与基于路径解构的图表示学习算法之间的关系，并基于此给出了在大规模图上 Node Conductance 的近似算法。在实验中，我们首先证明了 Node Conductance 相较已有节点中心度的独特性。又分别将其应用于静态图和动态图，发现其有极强的能力识别节点的活跃度。该工作已经被 PAKDD2020 接收。

第六章　图挖掘中路径解构策略和邻域解构策略的综合运用

本章介绍路径解构策略和邻域解构策略在图挖掘任务中的体现。在先前两章中分别介绍了两种策略的应用实例。邻域解构在刻画单一节点的任务中有重要意义，通过分析目标节点的邻域，可以剔除关于目标节点的噪音信息，丰富关于目标节点的有效信息。邻域解构的难点在于如何分析邻域，抽丝剥茧，筛选出邻域中有价值的部分。路径解构在刻画节点对连接紧密度的任务中有重要意义，利用“随机游走窗口内点对共同出现的频率”替代“点对间可行路径或最优路径的计数”，不仅大大简化了计算复杂度，同时也由于其与神经网络的训练过程十分契合，可以将深度学习的相关技术引入图挖掘领域。在本章中试图综合两种策略各自的优势，补全传统图挖掘算法中所忽视的信息，进一步提升算法的效果。

在本章的第一部分关于图的表示方法，关注的是节点的连续表示。本章提出一个节点信息辅助的图表示学习算法 SNS。图表示学习的目标是对图中节点的结构信息利用向量进行编码，使得节点之间在图中的关系可以通过向量距离计算快速得到。本文提出的算法 SNS 关注图中两种节点关系：节点之间的连接紧密性，节点地位的相似性，其中后者在传统的图表示学习算法中常常被忽略。SNS 算法利用路径解构策略完成对节点连接紧密程度的建模，利用邻域解构策略完成对节点地位的建模。与先前的邻域解构的实现方式不同，在 SNS 中，目标节点的邻域指的是目标节点 k-跳邻居组成的导出子图，而解构指的是该导出子图被分解为多个特殊的小子图。

在本章的第二部分关于图挖掘对具体应用，提出一个高效的节点地位表示方法，以支持跨结构洞节点的识别。SNS 算法中，基于邻域解构的节点地位刻画受限于计算复杂度过高，难以适用于大规模图。在本节提出的节点地位向量表示 RWSig 放弃了特殊小子图的计数操作，而使用随机游走路径替代特殊小子图，对目标节点的邻域进行路径解构，实现了邻域解构和路径解构的结合。

6.1　节点结构信息辅助的图表示学习 SNS

6.1.1　研究背景

基于图数据的机器学习模型有一个核心的问题：如何将图结构的信息与一般的机器学习模型相结合。举例来说，在边预测问题中，可以从每个点对的连接状况（如共同的朋友数等）和其是否有边相连的关系入手；在节点分类的任务中，可能会考虑是

以节点的地位（如边缘节点、核心节点、hub 类节点等）来分类，或者是以社区结构（图中相互连接紧密的一簇节点）来划分。然而，所有试图利用的这些信息都很难提取。没有一个高效的方法可以自动地从图类型的数据结构中提取信息，并把信息用特征向量表示出来。为了从图中提取丰富的结构信息，传统的方法一般利用的是图的统计信息（比如度数、聚集系数），核函数，或者通过特征工程选取图的局部连接特征。然而由于这些方法很大程度上都依赖人力，其使用范围都很有限。

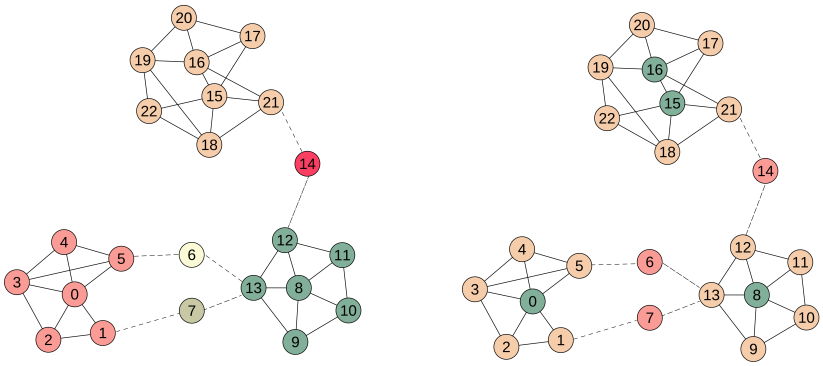
近年来出现了一个研究热点 — 设计方法去自动的学习图的结构特征，即图的表示学习。图的表示学习目标是将图中的节点或者子图映射到一个低维的隐空间中。在这个隐空间中，节点几何距离可以反映出它们在原图中的远近距离或者结构相似度。如果可以学出这样的一个隐空间，那么节点在隐空间中的向量表示就可以作为机器学习任务的输入数据。以往的工作往往将这个步骤视作预处理，通过人力挑选出来一些可能的结构特征。表示学习与之不同，它将这项工作本身也是作一个机器学习任务，利用数据驱动的方法对图结构信息进行编码。

DeepWalk[10] 首先将 word2vec 模型[170] 引入图表示学习领域。其作者开创性地将图中的随机游走路径和文档中的句子进行类比，图中的节点与文档中的单词进行类比。在这之后，有大量的改进性工作专注于对上下文（文本的上下文类比节点在图中的邻域）的定义，目标是刻画不同层次的图结构信息[39,40,45,67,68]。不过，这些改进的算法始终围绕这随机游走路径。这在一定程度上证明了，随机游走已经被研究者们认可，是一个可以刻画大规模图中节点之间关系的便捷方法。

尽管大部分的图表示学习算法都关注节点在图中的距离关系，试图将其保留在表示空间中，本文认为还有另外一种结构特征，值得表示学习算法保留：即节点的局部结构信息，对应着节点的不同“地位”。图6.1展示了两种节点的相似度，其中图6.1a所展示的邻接相似性指的是根据节点在图中的临近程度，并将相互连接紧密的节点归为一类。大部分的图表示学习算法都是基于这样的思想，在对节点进行分类时考虑的是节点邻居的类别。图6.1b展示的结构相似性则是另一种思路。结构相似性也称为 *structural equivalence*。可以根据不同的社区划分方式以及节点所处的位置将节点分为核心节点，边缘节点，桥节点。

以上两种节点相似度并不相互矛盾，相反，他们在具体的场景中是相互补充的，有各自适于的应用问题。例如，如果节点标签代表顾客的喜好，那么相较结构相似性而言，邻接相似性会更贴近顾客喜好的相似程度。如果节点标签代表社会地位，那么结构信息就至关重要了。对于一般的场景来说，一个理想的图表示学习方法应该在这两种相似性做出合理的平衡。

尽管结构相似性在最近的图表示学习相关工作中也被提及[66]，但是本文认为一般图表示学习所基于的随机游走模型是不具备刻画节点结构能力的。进一步来说，本文



(a) 邻接相似性　　(b) 结构相似性

GDV of node 0,2,8

		Orbit													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Node ID	*0*	0	5	5	0	0	0	0	0	0	0	6	0	3	1
	8	0	5	5	0	0	0	0	0	0	0	5	0	5	0
	2	4	1	2	0	2	0	0	0	2	2	0	2	1	0

$E(0,8) = 6^{0.5}$,　$E(0,2) = 98^{0.5}$

(c) 特殊子图度 GDV

图 6.1　在图 (a) 和 (b) 中, 两个被虚线连接的节点代表两点之间有很多中间节点。节点颜色代表它们所属的组。(a) 和 (b) 展示了划分节点的两个角度。在 (a) 中，相互连接紧密的节点被划分为一组。在 (b) 中，相同地位的节点被划为一类。(c) 展示的是节点 0,2,8 的特殊子图度向量 (GDV)。此处只展示了向量的前 14 维。$E(\cdot,\cdot)$ 是两个向量的欧式距离。

为了解决这一问题给出了基于特殊子图的刻画方法，使得新的图表示学习算法既能够捕捉节点之间距离的远近程度，也可以保留地位的相似性。

6.1.2　随机游走模型的局限性

传统的图表示学习试图抓取图中节点的空间临近性，并将这一关系保留在节点的向量隐空间中。本节从一个新的角度定义节点的相似性，即节点的局部结构相似性。两种方式的差异如图6.1所示。

DeepWalk[10] 的作者开创性地将图中节点的关系建模类比为自然语言中对单词的建模。在这种图表示学习的过程之中，节点对在随机游走路径的窗口中共同出现的次数被用于模型的训练。其中那些共现概率高的节点对会被认为在图中互为关系紧密的邻居。在模型的超参数数设定上，通常窗口大小会设定为一个大于 2 的值，因此可以认为这种邻居的关系是一种高阶的邻近性。

考虑一个由 N 个节点和 m 条边组成的图 G。对称邻接矩阵 $\boldsymbol{A}$ 中如果有元素的值为 1，则代表图中两点之间有边相连；如果为 0 则代表两点之间没有边。节点度数向量 $\boldsymbol{d} = \boldsymbol{A}\mathbf{1}$，其中 $\mathbf{1}$ 是一个大小为 $N \times 1$ 的全 1 向量。在图 G 上存在随机游走的路径，从

一点跳转至另一点时，跳转概率等于度数的倒数：

$$\boldsymbol{p}_{t+1} = \boldsymbol{p}_t \boldsymbol{D}^{-1} \boldsymbol{A} \equiv \boldsymbol{p}_t \boldsymbol{M}, \tag{6.1}$$

其中 $\boldsymbol{p}$ 是概率向量，$\boldsymbol{D}$ 是一个对角矩阵 $\boldsymbol{D} = \mathtt{diag}(\boldsymbol{d})$，$\boldsymbol{M}$ 是转移矩阵。令 $\pi = \pi \boldsymbol{M}$，可以求得随机游走达到稳定时的分布 $\pi = \boldsymbol{d}^{\top}/2m$, $P(i) = d_i/2m$。

对于一个从 i 节点出发的随机游走来说，在 r 步之后出现在节点 j 上的概率是：

$$P(j, r|i) = [\boldsymbol{M}^r]_{ij}, \tag{6.2}$$

$$P(i, j, r) = P(j, r|i)P(i) = \frac{d_i}{2m}[\boldsymbol{M}^r]_{ij} \propto [\boldsymbol{A}\boldsymbol{M}^{r-1}]_{ij}. \tag{6.3}$$

公式6.3证明两节点 i 和 j 在窗口内同时出现的概率是由两点之间的路径通过的节点决定的，而与 i 和 j 两节点本身毫无关系。很显然，路径中的节点边数越少，两个节点之间的短路径越多，两个节点出现在同一个窗口的概率越大。可以将这两种情况称为更直接的连接和更强的连接。图6.2展示了直接连接和强连接在图中的表现。当两个节点之间有很多短的可行路径时，两个节点之间的关系是强连接；当两个节点之间的中继节点度数很小时，两个节点之间的关系是直接连接。

在图表示的学习过程中，如果两个节点具备相似的邻居，那么在学习到的向量空间中，两个节点的位置也相似。而在算法实现中，邻居的范围实际上是由窗口大小来控制。在图中，如果两个节点的距离超过窗口大小，那么他们之间的相似性是无法被算法捕捉到的。换句话说，随机游走的采样方式限制了图表示学习算法只能刻画每个节点和其邻居之间的关系；而邻居的具体范围又与连接的强度和直接程度直接相关。

Grover et. al. 提出的 node2vec 算法[66]给随机游走引入两个参数 p 和 q，以控制游走在深度优先和宽度优先两种策略中的转换。倘如沿袭上文中对随机游走采样策略的分析，可以认为这两个参数实际上达到了修改转移矩阵 $\boldsymbol{M}$ 的效果。小的返回参数 p 倾向于两个具有强连接的节点互为邻居；小的转换参数 q 倾向于两个具有直接连接的节点互为邻居。因此，可以认为这个工作对图中邻居的定义进行了扩展，同时也需要指出其不足：（1）连接的强度和直接程度与节点的结构相似性是完全不同的概念（2）窗口以外的节点关系，该算法依旧无法捕捉。在接下来的实验中，也会通过可视化的方式来验证这一观点。

6.1.3 利用特殊子图刻画节点局部结构

根据上文的分析可知，传统的基于随机游走采样的图表示学习算法无法实现节点地位（局部结构）的刻画。为了到达上文提到的目标，需要一种更有效的节点地位刻画方式。

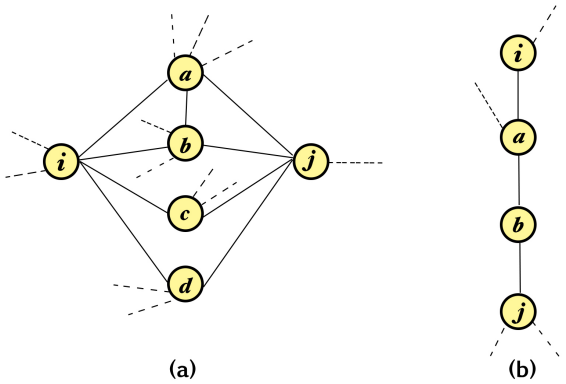


图 6.2　在两种情况下，i 和 j 出现在同一窗口的概率较大。一是二者之间是直接连接，二是二者之间是强连接。

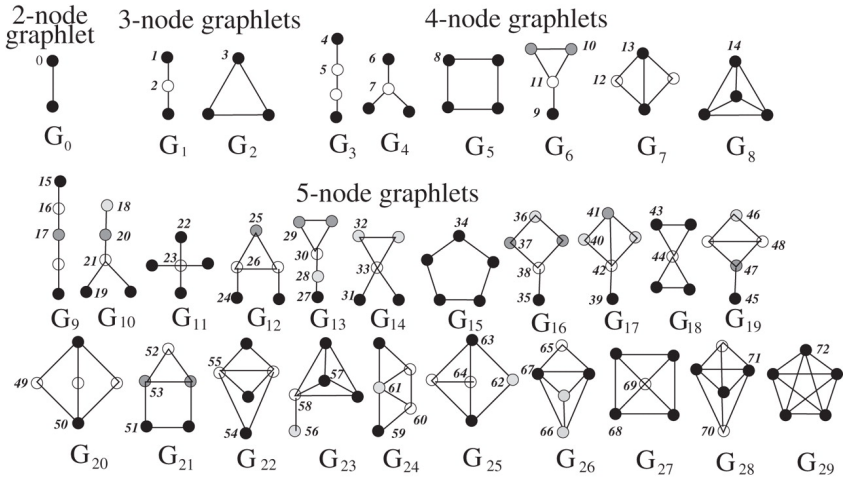


图 6.3　GDV 中涉及到的小子图[171]。相同颜色的节点属于同一个轨道。

其实，计算结构相似性在近几年也是一个受关注的工作。算法主要基于小子图 graphlets，树和随机游走[172]。本章的任务关注的是节点的局部结构特征，这在生物和社会科学中也有相应的研究[173]。下面介绍基于 graphlets 的结构识别及相似度计算。

graphlets 是一些互不同构的小子图。如图6.3展示，包含 2-5 个节点的小子图共有 30 个。轨道指的是 graphlets 中不同的节点位置，因此对称位置拥有相同的轨道编号，30 个小子图包含 73 个轨道。节点的 GDV *Graphlet Degree Vector* 可以视为将节点度数扩展为一个 73 维的向量。向量的每一维代表节点出现在某个轨道的次数。很显然出现在 0 号轨道的次数等于节点的度数。有了 GDV，可以通过 GDV 的欧式距离来表示节点结构的相似度。当然除了欧式距离以外，还可以通过其他方式，如斯皮尔曼相关系数，来计算相似性。在图6.1c中，计算了 0 号，2 号，和 8 号节点的 GDV（这是一个 14 位的向量，对应着三个节点和四个节点的小子图）。$E(\cdot,\cdot)$ 代表着两个节点 GDV 的欧式距离。根据欧式距离，可以清晰地得出结论：0 号和 8 号结构更为相近，同为核心位置的节点，而 2 号节点为边缘节点，0 号与 2 号之间的欧式距离远大于 0 号与 8 号之

间的距离。

graphlet 的枚举与计数也得到了广泛的研究。有大量的算法可以高效地在小图上对 graphlet 精确计数，在大图上对 graphlet 近似计数。在这里利用 *orca*[171] 算法完成该任务。

在应用 GDV 实现对节点所处局部结构的刻画和相似性计算时，探索出两个使用技巧使得算法的表现得到提升。

- **边缘轨道更重要。**在上文中，已经介绍了小子图 graphlet 和轨道 orbit 的概念。随着小子图包含的节点增加，轨道数，同时也是 GDV 的维数，会快速增长。而过多的特征会造成模型的过拟合。同时，在判断节点结构的相似性时，不同轨道所占的权重也应该是不同的。利用随机森林 Random Forest 对节点分类任务中轨道的重要程度进行分析，可以得到一些有趣的结论。

  随机森林包含许多棵决策树。在决策树中，每一个分支节点都代表某一种特征，基于这个特征可以对整个数据集进行切分。纯度用来衡量切分后的子集中节点标签的同质程度。对于节点标签分类任务来说，纯度通常利用基尼系数 Gini impurity 或者信息增益 information gain 来计算。当训练一棵决策树时，可以计算出每个特征为纯度带来的提升有多少。对于随机森林来说，同一个特征会出现在多棵树中，取该特征在所有数上的纯度提升均值为该特征的重要程度。值得注意的是，在这种计算方式下，如果一个特征对应着多个类别或者有极少的相关特征时，该特征会被认为是重要的。

  利用随机森林来计算轨道 I_o 的重要性，该实验在多个数据集上都重复。表6.1展示的是在 BlogCatalog 数据中，不同轨道的相对重要性 RI_o。在其他的数据上也会得到相似的发现。

$$RI_o = \frac{I_o}{\max\limits_{0 \leqslant i \leqslant 72} (I_i)} \times 100\%$$

  可以发现 0 号轨道（也就是节点度数）在节点分类任务上是最不重要的特征。从表中可以观察到在一个 graphlet 中，那些边缘的轨道（该轨道位置所对应的度数很小）比核心轨道更重要（该轨道位置所对应的度数很大）。比如说，在 G_1 中，轨道 1 比轨道 2 更重要；在 G_{12} 中，轨道 24 最重要，而轨道 26 最不重要。针对这一有趣的现象，此处尝试给出一些解释。GDV 的每一维之间实际上存在着一些依赖关系。以一个至少有两个邻居的节点为例，它可以与任意两个邻居组成 G_1 或者 G_2。如果将 C_i 代表轨道 i 的 graphlet degree，可以得到：

$$\binom{C_0}{2} = C_2 + C_3. \tag{6.4}$$

  也就是说，核心轨道的 graphlet degree 更有可能是冗余的，它可以由一些边缘轨

表 6.1　BlogCatalog 数据集的节点分类任务中，不同轨道的相对重要性 RI_o。

orbit	RI%	orbit	RI%	orbit	RI%	orbit	RI%	orbit	RI%
0	0.05	1	21.18	2	1.79	3	0.14	4	84.58
5	62.26	6	95.38	7	10.94	8	21.27	9	81.94
10	63.48	11	8.50	12	41.98	13	6.15	14	3.71
15	92.27	16	76.27	17	85.71	18	86.70	19	93.90
20	82.76	21	62.11	22	86.08	23	20.55	24	86.28
25	83.70	26	65.51	27	88.68	28	81.02	29	82.77
30	63.89	31	100.00	32	76.21	33	20.14	34	75.33
35	95.19	36	77.29	37	82.30	38	44.18	39	94.31
40	77.64	41	65.50	42	18.75	43	80.58	44	11.52
45	87.21	46	80.75	47	58.46	48	65.70	49	83.55
50	24.42	51	76.65	52	80.50	53	43.72	54	85.05
55	15.38	56	91.93	57	66.20	58	14.91	59	77.65
60	55.84	61	13.73	62	74.85	63	31.37	64	41.12
65	83.61	66	59.60	67	16.18	68	39.53	69	7.20
70	49.11	71	14.99	72	12.14				

道的值求和计算出来。综上所述，在基于 GDV 计算两个节点之间的结构相似度时，应该将更多的权重赋予边缘轨道。

- **限制相似节点的个数。**为了提升图表示学习的效果，本章提出的算法尝试在学习的过程中引入节点的局部结构特征。对于一个目标节点来说，其表示学习的过程会涉及到基于距离的邻居节点以及基于结构的相似节点。其中结构相似的节点可以从整图中筛选，也可以从目标节点的邻域中选取。在某些情形下，节点之间的距离与分类标签更加相关，结构相似的节点从目标节点的近邻中挑选更有意义。在另外一些情形中，节点的局部地位对标签分类起到决定性意义，那么在图的全局范围内选取结构相似的节点是必要的。

在算法的预处理阶段，首先利用统计 graphlet 数量的算法为每个节点产生一个 GDV。其次，针对每个节点，找到 GDV 空间中 cosine 相似度最大的 K 个节点。其中在搜索相似节点时，两个节点之间的距离需要被限定在 S 步之内：$\mathcal{N}_{v_i}^S$ as:

$$\mathcal{N}_{v_i}^S = \{v_j \mid \boldsymbol{A}_{ij} = 1 \vee \boldsymbol{A}_{ij}^2 \geqslant 1 \vee \cdots \vee \boldsymbol{A}_{ij}^S \geqslant 1\}.$$

搜索出的结构相似节点对会以稀疏矩阵的形式保存在 $\boldsymbol{S}$ 中，其中 s_{ij} 不为 0 时代表 j 节点是 i 节点的 K 最近邻之一，注意 $s_{ij} \in [0, 1)$ 且 $\sum_j s_{ij} = 1$。也就是说，$\boldsymbol{S}$ 的每一行中有 K 个非零项。总的来讲，预处理部分涉及 5 个可以调整的参数：

1. 衡量 GDV 相似度的指标
2. O: 参与计算的轨道数

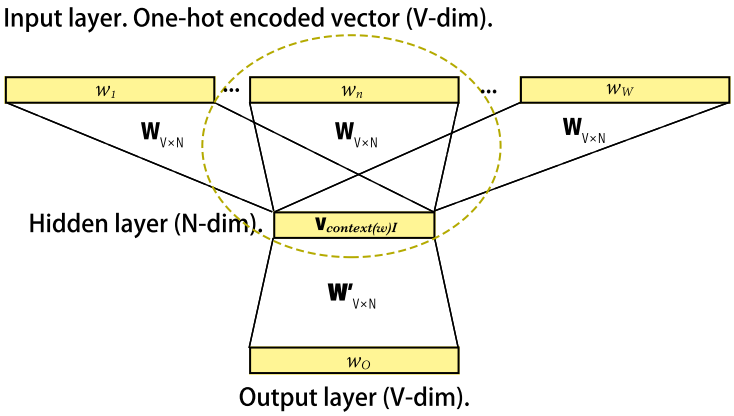


图 6.4　CBOW 算法框架。$w_1, w_2, \cdots, w_W$ 是目标词的上下文。词表的大小是 V。窗口大小是 W。$\boldsymbol{W}$ 是输入层和隐层之间的权重矩阵。$\boldsymbol{W}$ 的每一行是一个 N 维的词向量 $\boldsymbol{v}_w$。类似的，$\boldsymbol{v}'_w$ 是 $\boldsymbol{W}'$ 中的行向量。

3. R: 计算相似度时不同轨道的权重
4. K: 截取前 K 个结构最相似的节点
5. S: 所选取的结构相似节点之间最大距离阈值

6.1.4　模型描述

本节提出了一种同时利用节点邻接距离和结构相似性的图表示学习方法。

6.1.4.1　基于负采样的 CBOW 算法

尽管本节提出的方法可以适用于增强任何基于 word2vec 的图表示学习算法，在此仅以 CBOW[170] 为例介绍本节的方法与经典图表示学习的融合方式。与 Skip-gram 相比，CBOW 算法在短句子和大规模的数据集上有着更好的表现。

在自然语言处理的研究中，通常任务相似的词也有着相似的上下文。CBOW 即是用一个词的上下文来预测该词。CBOW 方法的损失函数如下，其中 C 是词表，w 是要预测的词。

$$\mathcal{L} = \sum_{w \in C} \log p(w|\texttt{Context}(w)). \tag{6.5}$$

CBOW 以上下文单词的 one-hot 编码作为输入 w_I，以输入单词的平均词向量作为隐层，其中 W 是上下文 w 的个数，$\boldsymbol{v}_{w_i}$ 是 w_i 的词向量。

$$\boldsymbol{h} = \boldsymbol{v}_{w_I} = \frac{1}{W}(\boldsymbol{v}_{w_1} + \boldsymbol{v}_{w_2} + \ldots + \boldsymbol{v}_{w_W}). \tag{6.6}$$

w_O 是 CBOW 的输出，y_j 是输出层的第 j 个元素。$\boldsymbol{v}_w$ 和 $\boldsymbol{v}'_w$ 是单词 w 的输入和输出向

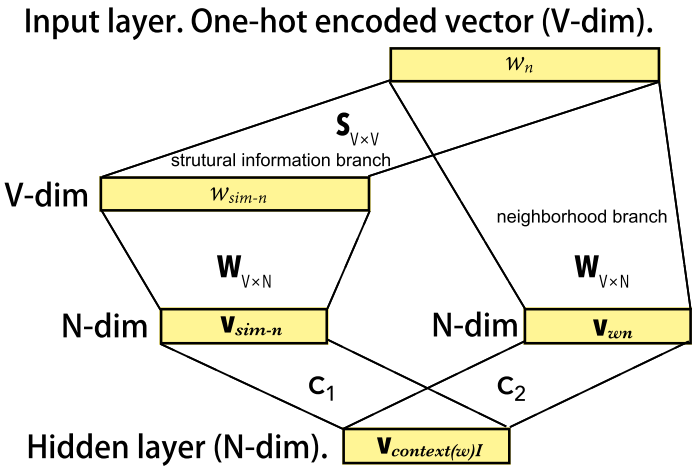


图 6.5　将结构信息与图表示学习的过程相结合。这里只展示单词 w_n 的学习过程。

量。

$$p(w|\texttt{Context}(w)) = y_j = \frac{\exp(\boldsymbol{v}_{w_j}^{'\top}\boldsymbol{v}_{w_I})}{\sum_{j'=1}^{V}(\boldsymbol{v}_{w_j}^{'\top}\boldsymbol{v}_{w_I})}. \tag{6.7}$$

softmax 因为分母涉及到的计算量很大，方程很难优化。一种有效的近似方式是负采样的方法。正如论文[170]中所述，负样例从词频的 $\frac{3}{4}th$ 幂次分布中均匀采样得到。将采样的分布记做 $P_n(w)$，σ 是逻辑斯蒂方程。$\mathcal{W}_{neg}$ 是 k 个负样例组成的集合，该集合从 $P_n(w)$ 中抽取得到。

$$P_n(w) = \frac{U(w)^{\frac{3}{4}}}{Z}, \tag{6.8}$$

$$\mathcal{L} = \log\sigma(\boldsymbol{v}_{w_O}^{'\top}\boldsymbol{v}_{w_I}) + \sum_{w_j\in\mathcal{W}_{neg}} \log\sigma(-\boldsymbol{v}_{w_j}^{'\top}\boldsymbol{v}_{w_I}). \tag{6.9}$$

6.1.4.2　图表示学习中的结构分支

本节提出了一个新的图表示学习框架，综合邻接信息和结构信息。图 6.4展示了 CBOW 模型。SNS 对图6.4中虚线内部分的结构进行了重构，重构后的结构如图6.5所示。在 SNS 模型中，目标点不仅仅由其周围的节点，也由其结构相似的节点推断出来。通过 GDV 可以计算出每一对儿节点之间的相似度，得到相似度矩阵 $\boldsymbol{S}$，它的每一行包含 K 个非零值，是 K 个与目标节点结构最相近的相似度。邻居分支和结构分支共享输入的节点表示矩阵 $\boldsymbol{W}$。$\boldsymbol{W}$ 矩阵中的每一行是节点的向量表示。节点的结构向量表示如下：

$$\boldsymbol{v}_{sim\text{-}n} = \sum_{m=1}^{V} s_{nm}\boldsymbol{v}_{w_m}. \tag{6.10}$$

相似度 s_{nm} 可以视作赋予 w_m 的权重，m 是与 n 节点结构最相似的 K 个节点。

最终的节点的向量表示将两部分融合在一起：

$$\boldsymbol{v}_{w_I} = \frac{1}{W}\sum_{n=1}^{W}(c_1(\mathsf{deg}(w_n))\boldsymbol{v}_{w_n} + c_2(\mathsf{deg}(w_n))\boldsymbol{v}_{sim\text{-}n}), \tag{6.11}$$

其中 c_1 和 c_2 是将两者结合在一起的比例参数 $(c_1, c_2 \in (0,1))$。通常，度数高的节点可以通过邻居获得足够多的信息。而度数低的节点则需要结构相似性来补充邻居的不足。因此，度数高时，c_1 的值大于 c_2，反之亦然。SNS 将节点按照度数排序分成 C 段，同一段中的节点共享相同的 c_1，c_2 值。

总的来说，SNS 模型对节点的邻居信息和结构信息进行加权求和，使得最终的图表示学习结果包含两部分信息。

6.1.4.3 学习过程

矩阵 $\boldsymbol{S}, \boldsymbol{W}, \boldsymbol{W}'$ 和参数 c_1, c_2 的值都通过反向传播来更新。其中 $\mathcal{L}$ 对 w_j 的偏导数计算如下：

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}_{w_j}^{'\top}\boldsymbol{v}_{w_I}} &= \begin{cases} \sigma(\boldsymbol{v}_{w_j}^{'\top}\boldsymbol{v}_{w_I}) - 1, & \text{if } w_j = w_O; \\ \sigma(\boldsymbol{v}_{w_j}^{'\top}\boldsymbol{v}_{w_I}), & \text{if } w_j \in \mathcal{W}_{neg}. \end{cases} \\ &= \sigma(\boldsymbol{v}_{w_O}^{'\top}\boldsymbol{v}_{w_I}) - t_j. \end{aligned} \tag{6.12}$$

其中 t_j 是正负样本的标识. 根据链式法则，可以进一步得到：

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}'_{w_j}} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{v}_{w_j}^{'\top}\boldsymbol{v}_{w_I}} \cdot \frac{\partial \boldsymbol{v}_{w_j}^{'\top}\boldsymbol{v}_{w_I}}{\partial \boldsymbol{v}'_{w_j}} = (\sigma(\boldsymbol{v}_{w_O}^{'\top}\boldsymbol{v}_{w_I}) - t_j)\boldsymbol{v}_{w_I}, \tag{6.13}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}_{w_I}} &= \sum_{w_j \in \{w_O\}\cup\mathcal{W}_{neg}} \frac{\partial \mathcal{L}}{\partial \boldsymbol{v}_{w_j}^{'\top}\boldsymbol{v}_{w_I}} \cdot \frac{\partial \boldsymbol{v}_{w_j}^{'\top}\boldsymbol{v}_{w_I}}{\partial \boldsymbol{v}_{w_I}} \\ &= \sum_{w_j \in \{w_O\}\cup\mathcal{W}_{neg}} (\sigma(\boldsymbol{v}_{w_O}^{'\top}\boldsymbol{v}_{w_I}) - t_j)\boldsymbol{v}'_{w_j} \equiv \Delta. \end{aligned} \tag{6.14}$$

根据以上计算，可以得到如下的迭代更新公式：

$$\boldsymbol{v}_{w_j}^{'(new)} = \boldsymbol{v}_{w_j}^{'(old)} - \eta(\sigma(\boldsymbol{v}_{w_O}^{'\top}\boldsymbol{v}_{w_I}) - t_j)\boldsymbol{v}_{w_I}, \tag{6.15}$$

$$c_1^{(new)} = c_1^{(old)} - \eta\boldsymbol{v}_{w_i}\Delta, \;\; c_2^{(new)} = c_2^{(old)} - \eta\boldsymbol{v}_{sim\text{-}i}\Delta, \tag{6.16}$$

$$\boldsymbol{v}_{w_i}^{(new)} = \boldsymbol{v}_{w_i}^{(old)} - \eta c_1\Delta, \;\; \boldsymbol{v}_{w_j}^{(new)} = \boldsymbol{v}_{w_j}^{(old)} - \eta c_2 s_{ij}\Delta, \tag{6.17}$$

$$s_{ij}^{(new)} = s_{ij}^{(old)} - \eta \boldsymbol{v}_{w_j} \Delta. \tag{6.18}$$

注意到矩阵 $\boldsymbol{W}$ 在每一轮迭代中实际上被更新了两次，一次是在邻域分支中被更新，另一次是在结构分支中被更新。当迭代结束，将 $\boldsymbol{W}$ 做为最终的节点向量表示结果。

6.1.4.4　算法变种

在此简要讨论一下 SNS 模型可能存在的其他变种形式。针对不同的任务和数据集，可以选择一个适当的形式。

- 以哪个节点为基准选取结构相似节点？
  在上文介绍的框架中，模型在每轮计算中选取的是上下文节点的结构相似节点参与计算，而不是目标节点的结构相似节点。实际上，选取目标节点的结构相似节点也是可行的。这两种方案的差别在于选取相似节点的范围是不同的。如果希望在一个更大的范围内（与目标节点的最大距离大于 3）搜寻结构相似的节点，那么后一种方案中，预处理阶段会承受极大的计算压力。相反，在现行方案中，可以通过增大窗口 $\mathcal{W}$ 以轻松实现这一目标。大的窗口大小不会明显地影响算法效率。不过，现行方案的缺点也是存在的：在很多时候节点之间的相似性并不是可传递的（与上下文节点相似的节点与目标节点不一定相似），而可传递性与图的同配性有很大关系（图中相似的节点更有可能互相连接）。可以根据具体的数据集和应用场景考虑选择其中一个方案。
- 将参数 $\boldsymbol{c_1}, \boldsymbol{c_2}$ 和相似度矩阵的值固定。
  在上文介绍的框架中，参数 $\boldsymbol{c_1}, \boldsymbol{c_2}$ 和相似度矩阵的值是在学习过程中被更新的。发现给这些参数一个高质量的初始值，并且在训练中固定住也可以取得不错的效果。并且，固定参数还可以加速整个学习过程。在实验中，设定初始值的技巧是给度数高的节点设定较高的 c_1 和较低的 c_2 值。如果 j 是 i 节点的 K 个结构相似节点之一，则设定 $s_{ij} = 1/K$，否则 $s_{ij} = 0$。

6.1.5　实验评测

在本章中，首先展示一些图表示学习可视化的效果，以证明 DeepWalk，node2vec 和本章提出的算法之间的表示能力差异。随后展示几个图表示学习算法在节点多标签分类任务上的效果。最后将会谈论参数敏感性的问题，并给出结构相似信息和邻域信息在实际应用的一些建议。

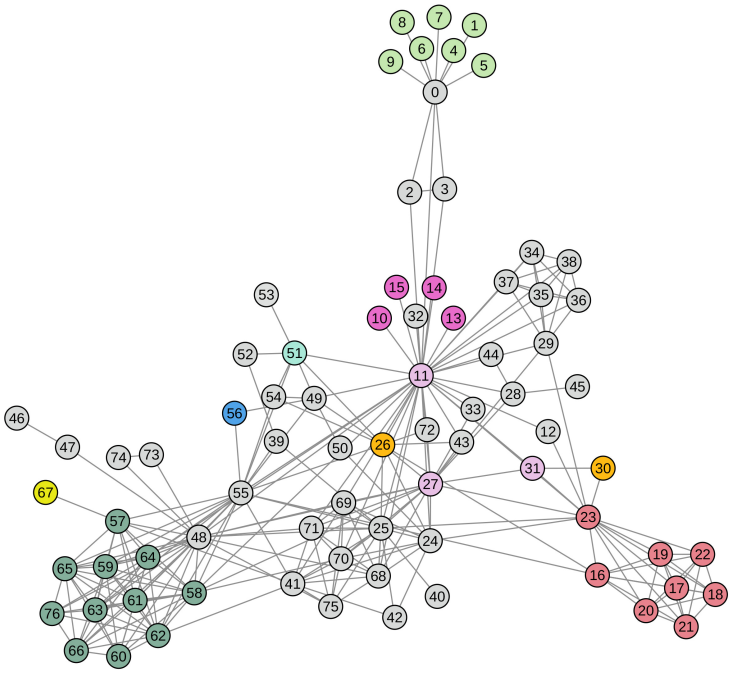


图 6.6　Les Miserables 角色共同出现网络。节点的布局利用原子引力斥力法: ForceAtlas2[174]。

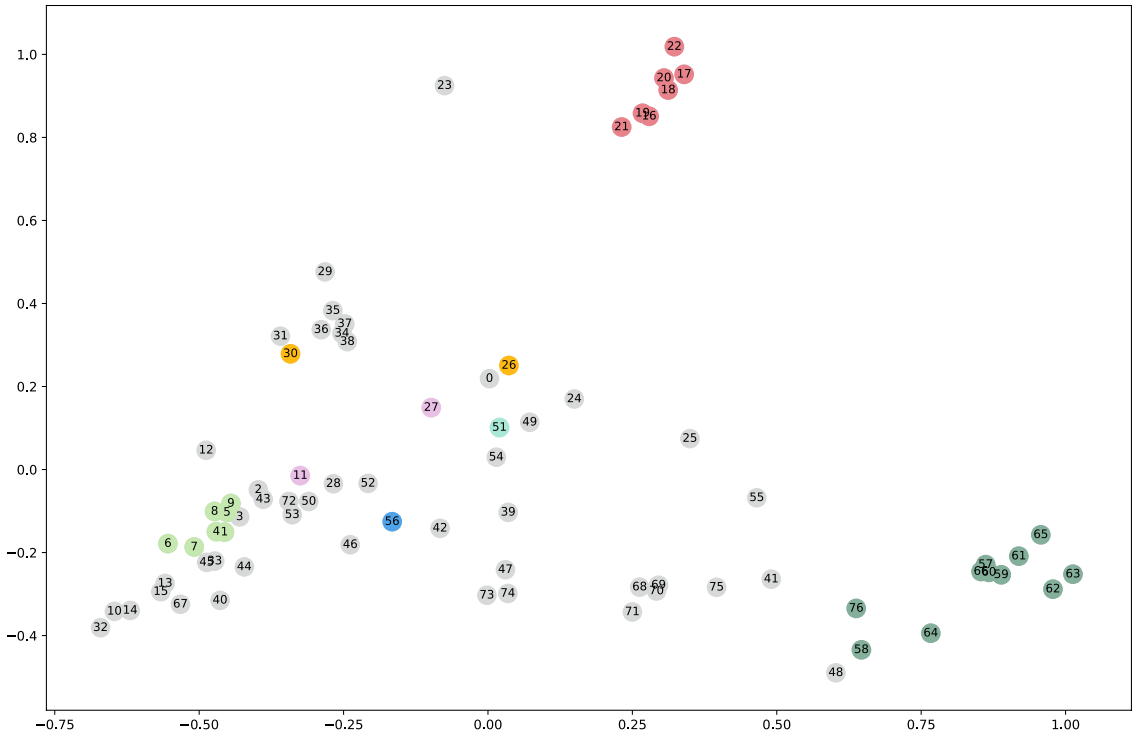


图 6.7　利用 DeepWalk[10] 来学习 Les Miserables 图的节点表示 ($p = 1, q = 1, d = 16$)。节点向量通过 PCA 算法降维到二维空间。

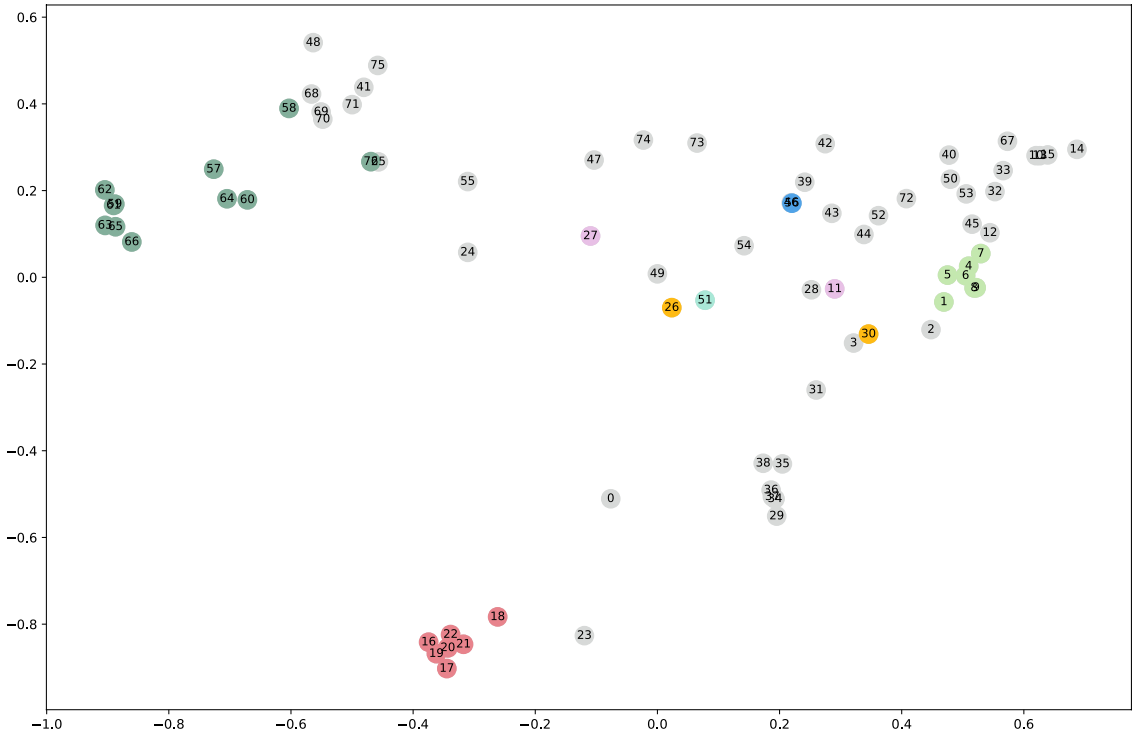


图 6.8　利用 node2vec[66] 来学习 Les Miserables 图的节点表示 ($p = 1, q = 2, d = 16$)。节点向量通过 PCA 算法降维到二维空间。

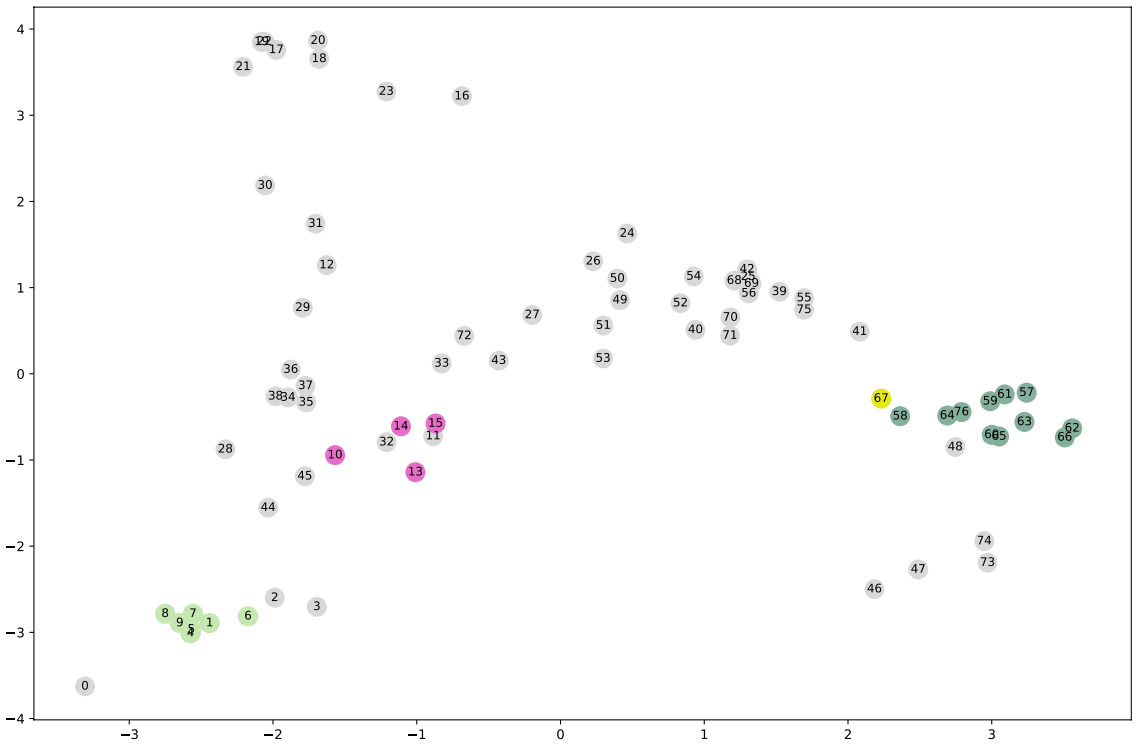


图 6.9　利用 SNS 来学习 Les Miserables 图的节点表示。节点向量通过 PCA 算法降维到二维空间。

6.1.5.1 案例分析

Les Miserables 图包含 77 个节点和 254 条边，每个节点代表故事中的任务，边代表他们在同一事件中出现。对比算法包括 DeepWalk[10]和 node2vec[66]算法。他们都是典型的基于随机游走的图表示学习算法。node2vec 算法声称通过调节参数 p 和 q，算法有能力为具有相似子结构的节点学习到相似的向量表示。本实验沿用了他们在论文中汇报的参数 ($p = 1, q = 2$)。DeepWalk 实际上是 node2vec 的一种特殊情况（$p = 1$，$q = 1$）。简便起见，本实验将窗的尺寸设置为 4，也就是说目标节点与其邻域节点的最大距离为 2。

图6.6是 Les Miserables 图，布局采用了 ForceAtlas2[174]，这是一个与原子之间力的相互作用相关的布局方式。图6.7，图6.8和图6.9分别是 DeepWalk，node2vec 和 SNS 学习到的向量表示。向量的维数被设定为 16，实验利用 PCA 算法将 16 维降至 2 维。在图6.6-6.9之中，相同的节点对应着相同的颜色，方便在多幅图中找到对应的节点。

在上面的章节中得到的结论是基于随机游走的图表示学习对节点之间的直连接和强连接比较敏感。在图6.6中，51 和 56 号都是 26 的邻居。然而 26 和 51 之间有很多短路径，26-51，26-49-51，26-54-51，26-11-51 等等。然而 26 和 56 之间的路径很少：26-49-56 和 26-55-56。也就是说，26 号点和 51 号点之间的关系更强。在图6.7中可以观察到，26 号节点和 51 号节点之间的距离更近。在图6.6中，11 号节点，16-22 号节点和 27 号节点都是是 30 号节点的二步邻居。处在中间位置的 23 号节点和 31 号节点之间有着更多的边。相对应的在图6.7中，节点 16-22 离 30 号节点的距离更远。

在随机游走中对参数 p 和 q 的控制实际上是对更直接的连接或者更强的连接两者之间做了偏向。在实验中，node2vec 的参数设置为 $p = 1, q = 2$，因此它对强连接更加敏感。延续对于 DeepWalk 的讨论，26 和 51 之间的关系相较与 26 和 56 之间的关系更加强。在图6.8中可以发现，26 和 51 十分接近，远远小于他们在图6.7的距离。

在图6.6中，1 号节点，4-9 号节点有着相同的局部结构。他们都有一条边连接至 0 号节点。57-65, 76 节点同样也是一群局部结构类似的节点，它们共同形成了一个完全子图。在图6.7和图6.8中发现两类节点的位置特点完全不一致。57-65, 76 节点完全分散，然而 1, 4-9 聚集在一起无法分开。这一差别证明无论随机游走的参数如何变化，它都无法捕捉节点的结构相似性。57-65, 76 节点远离其他节点的原因在于这几个点之间的连结既强又直接，以至于从他们出发的随机游走鲜有机会访问到其他节点。与之形成对比的是，1 号和 4-9 号节点，他们之间不具备这种类型的局部结构。基于随机游走采样的图表示学习算法是无法捕捉这类节点的结构相似性的。

在图6.9中，这两类节点都聚集在一起。这一提高要归功于对于特殊子图的使用。注意到尽管 10, 13-15 同样也有一条边，但他们与 1, 4-9 号节点的局部位置仍是完全不

同的。例如，67 号和 57 号在图6.6中相连，同时 57-65 和 76 形成一个完全子图。在这种状况下，可以认为 67 与 57-65 和 76 更像。13-15 号节点都和 11 号节点有一条边，可以认为 10, 13-15 和 1, 4-9 更相像。这些关系在图6.9中都能得到反映。

以上的一系列展示分析试图证明随机游走采样法不善于挖掘节点之间的结构相似性。随机游走自身的特性限制了这类算法的能力。他们只能够捕捉节点之间连结的强度和直接性，而非节点自身的结构特征。本章提出的 SNS 算法，借助特殊子图分布向量 GDV，更擅长完成节点结构特征的刻画。

6.1.5.2　实验准备

数据集

在接下来的实验中，本文选取了一些与出发点相符合的几个数据集。这些图中节点的标签分布都呈现出了同质性（相互紧密连结的节点标签一致）和同构性（具有相似局部结构的节点其标签也相似）[66]。

1. **BlogCatalog**[175] 是一个博客社交网站，用户可以在该网站发布博客，关注好友等。该图数据集的节点代表用户，节点标签代表用户的兴趣，边代表用户之间的联系。在这个社交图中，相似爱好的用户更有可能相互关注成为朋友。同时，有相似结构特征的用户即便不熟识，也有可能具有相同标签（例如都是意见领袖）。该图包含 10312 个节点，33983 条边和 39 个标签。
2. **Protein-Protein Interactions**[176] 是一个人体内生化反应的蛋白质交互图，标签代表蛋白质的生物学属性。该图包含 3890 个节点，76584 条边和 50 个标签。
3. **POS**[177] 是一个 Wikipedia 语料中单词的共现图。POS 是单词的标签，代表词在语法方面的属性。在文本中，经常共同出现的单词有更大的概率代表同一语义；同时，在文本中，那些拥有相似图结构的单词有更大可能是同一类型的词（比如“苹果”和“梨”）。该图包含 4777 个节点，184812 条边和 40 个标签。

对比算法

本实验选取了一个经典的降维算法和三个具有代表性的图表示学习算法做为对比算法。

1. **Spectral Clustering**[178] 是一个基于矩阵分解和最小化标准化割（Normalized Cut）的降维算法。与原论文中的保持一致[178]，本实验将维度 d 设定为 500。
2. **DeepWalk**[10] 是第一个将深度学习技术引入图表示学习的模型。它是一个无监督的算法，利用随机游走来采样节点的邻域，并使用一个带负采样的 skip-gram 模型来学习。与原论文中的保持一致[10]，本实验将模型所涉及的参数设定为 $d = 128, r = 10, l = 80, k = 10$
3. **LINE**[39] 定义了一个与节点一度邻居和二度邻居直接相关的优化目标。模型会

分别基于这两部分信息学习出维度为 $d/2$ 的表示向量，最终再将两部分直接拼接做为模型输出。在监督学习的模式下，该模型可以动态的调整两部分信息所占的比重，并有可能取得更好的表示效果。为了公平起见，在本文的实验中，选取无监督学习的模型，并与原论文保持一致[39]，将模型所涉及的参数设定为 $d = 128, r = 10, l = 80, k = 10$。

4. **node2vec**[66] 利用受到参数控制的游走对图结构进行采样。最重要的两个参数 p 和 q 会控制游走在深度优先策略和宽度优先策略之间摇摆。DeepWalk 实际上是 node2vec 模型的一个特例，其 $p = q = 1$。node2vec 是一个半监督的算法，它需要 10% 的标签数据来确定一个比较合适的 p 和 q 值。本实验依旧与原[66]论文的参数设定保持一致。

本章节中提出的算法 SNS 设定如下。在预处理部分，$K = 5, S = 1, O = 14, R = 9$。在学习部分和 DeepWalk 模型选取同样的参数，$C = 5$。预处理部分涉及到特殊小子图的计数，本实验使用 *orca*[171] 算法。根据该算法的论文，在台式机上 (Intel Core 2, 2.67 GHz)，一个包含 25368 节点和 75004 条边的图，*orca* 仅需 2.5s 即可完成 4 节点特殊子图的统计工作。在耗时方面，SNS 相比 DeepWalk 会消耗更多的时间，原因在于 SNS 有更多的参数需要优化。与这两个方法相比，node2vec 算法在游走采样阶段需要消耗更多的时间。

6.1.5.3 多标签分类任务

在这个任务中选取了和 node2vec 论文[66] 中一样的数据集和实验过程。为了完成标签分类的任务，习得的节点向量表示会做为一个 logistic 回归模型的输入。本实验选取一部分标注节点做为分类模型的训练集，剩下的部分做为测试集。整个过程会重复 10 次，在图6.10中展示的是 10 次实验的平均 Micro-F1 值和平均 Macro-F1 值。

可以清楚地发觉在三个数据集上，SNS 相较其他对比算法都有一定的优异。Spectural Clustering 仅仅在 BlogCatalog 数据上比较有竞争力。DeepWalk 在多个数据集上表现最为稳定。这说明，不同的的邻域采样策略在不同的应用领域中并不总是可依赖的。涉及一些特殊的采样策略往往会限制模型在不同领域里的表现。在 BlogCatalog 数据集上，SNS 的优势并不如另外两个数据集那么明显。所有神经网络相关的算法都可以取得相近的分数。推测 BlogCatalog 数据集中，节点标签与节点邻域更加相关，而与节点地位没有太强的相关性。相反，在 PPI 数据集和 POS 数据集上，SNS 的表现可以归功于对节点地位的建模。

图6.11展示的是在 BlogCatalog 上，算法参数敏感度的实验。在其他两个数据上也做了相同的实验，并得到相似的结果。图6.11的前两个子图关注的是与结构相似度相关的参数。O 代表参与计算的轨道数。R 代表计算相似度时不同轨道的权重。S 代表所

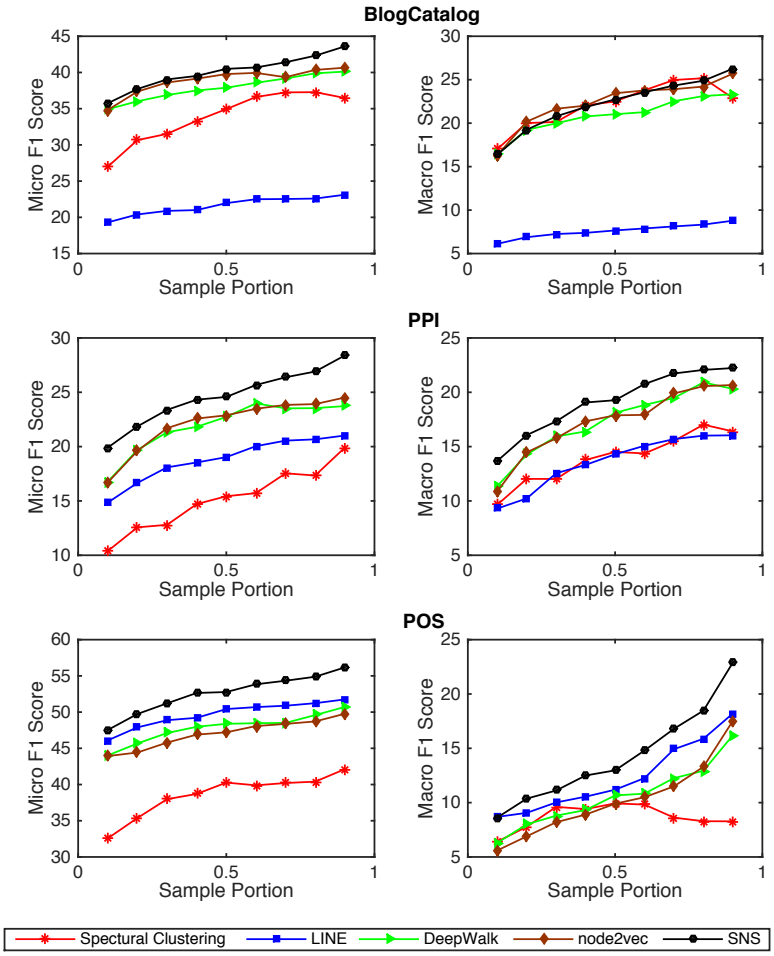


图 6.10　数据中训练集的比例变化时，算法结果的 Macro-F1 分数和 Micro-F1 分数。

选取的结构相似节点之间最大距离阈值。如果节点标签与节点在图上的距离相关，那么较大的 S 值会导致远距离的结构相似节点被错误地加入学习过程，使得最终的标签分类任务表现不佳。在6.1.3章中讨论的轨道重要性在本实验中亦有证明。如果边缘节点的权重大于核心节点，那么效果可以得到提升。轨道数目不会对表示向量的质量造成很大影响。考虑更多的轨道并不能提升分类的效果。2 节点，3 节点和 4 节点特殊子图所对应的总共 14 个轨道已经足够。

剩下的几个子图是关于随机游走阶段的几个模型参数。窗口大小 k 对结果的影响较小。在窗口较小时模型依旧可以取得较好成绩的原因在于 SNS 模型引入了窗口外部的结构相似节点。相反，比较大的窗口值反而会给模型训练过程中带来过多噪音，影响表示向量的质量。每个节点的随机游走数目 r 和路径长度 l 相对而言对结果的影响比较大。这两个参数共同控制了训练数据的规模。增加表示向量的维度 d 可以有效提升分类效果。当维数大于 100 时，提升变得不是很明显。

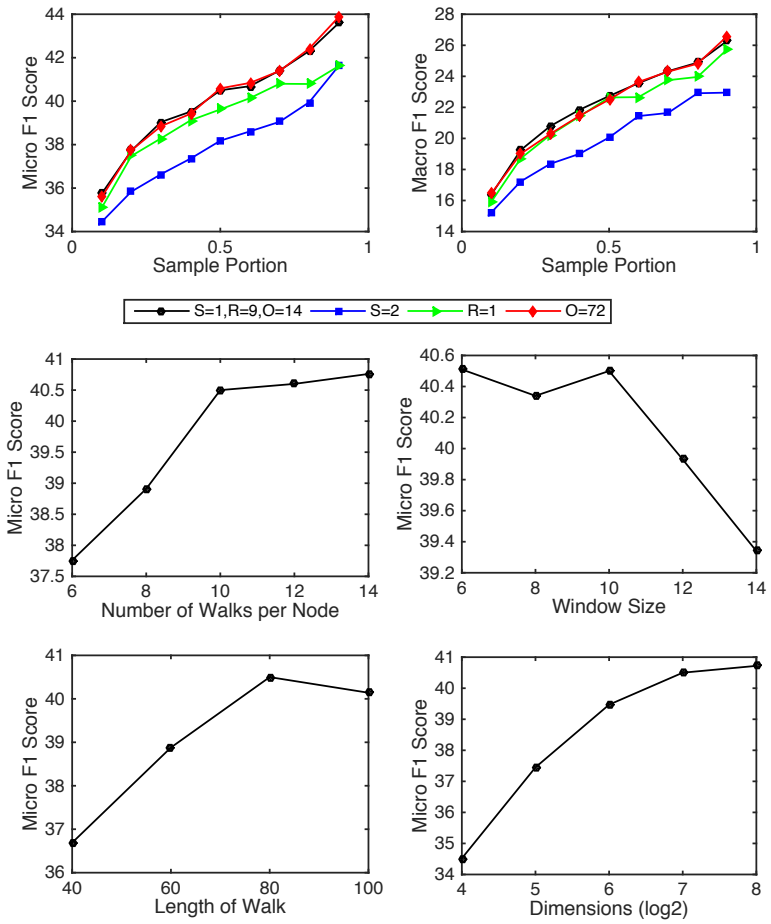


图 6.11　BlogCatalog 数据集上 SNS 算法的参数敏感度。

6.2　识别跨结构洞节点

6.2.1　应用背景

图表示学习作为一种有效的工具，极大地提升了了节点分类、边预测等等任务上的表现。除了希望通过图表示学习捕捉节点丰富的邻接状态以外，近年来还有一些研究希望对节点的局部结构做出向量化的刻画。然而这些方法都受限于计算复杂度，无法应用于大规模网络。

struc2vec 利用邻域节点的度序列来表示每个节点所处的结构特征，这一操作的复杂度为 $O(n^2)$。上一章节提出的 SNS 模型利用特殊子图度向量来表示结构特征，但是特殊子图的计数工作本身也需要消耗大量计算资源。GraphWave 模型[179]依赖于拉普拉斯矩阵的特征分解，当邻接矩阵的规模很大时，特征分解是很难避免的计算瓶颈。

本章提出来一种新的建模节点局部特征结构的方法 RWSig，将节点所处的局部子图编码为一个向量。两个节点所处结构的相似性可以通过向量的欧式距离计算得到，

而与两个节点所处的位置远近无关。RWSig 与图谱之间存在着潜在的关系，这也从另一个角度证明了 RWSig 的有效性。本章将经典的图表示学习得到的向量表示与 RWSig 拼接在一起，得到的新向量即包含了节点的邻接特性，又包含了节点的结构相似性。新向量在识别跨结构洞节点的任务上相较传统算法有着极大的效果提升。同时，RWSig 有着更低的复杂度，可以轻松处理大规模网络。

6.2.2　基于随机游走的节点结构向量 RWSig

RWSig 的主要目标是将一个导出子图 G_i 编码为一个定长的向量，其中 G_i 是节点 i 的 k 步邻居共同组成的子图。在 RWSig 中，节点 i 利用在随机游走中和其他节点出现在同一窗口内的次数，以其为特征其转换为向量。给定窗口大小 w，路径长度 l 和由每个节点起始的路径数目 p，总共有 $p \cdot n$ 条路径被生成，其中 n 是节点个数。RWSig 用一个长度为 $2w$ 的滑动窗口处理每条路径，记录同时出现在窗口内部的点对次数。这一过程与经典的图表示学习算法 DeepWalk 是完全一致的[10]。

处理完每条路径后，RWSig 为每个节点生成一个向量 $\boldsymbol{v} \in \mathbb{R}^n$，其中 n 是节点个数。假设节点的编号是连续的，处在 0 到 $n-1$ 之间，对于某个节点 a 来说，向量 v_{a_i} 第 i^{th} 位的数值即为 i 号节点出现在以 a 为中心的窗口内的次数。RWSig 将向量做归一化处理：

$$\texttt{normalize}(v_{a_i}) = \frac{v_{a_i}}{\sum_n v_{a_n}}. \tag{6.19}$$

归一化后，向量 v_{a_i} 第 i^{th} 位表示 i 号节点出现在以 a 为中心的窗口内的概率。

其实很容易想到如果两个节点 m 和 n 有着很相似的结构，那么他们对应的向量应该有着相似的数值组成。换句话说，刻画节点的局部结构与刻画归一化后向量的数值分布是等价的。因此 RWSig 利用累积量方程对向量进行转换[180]。

对于一个随机变量 X 来说，累积量方程定义如下：

$$K_X(t) = \log \mathbb{E}[e^{tX}], \tag{6.20}$$

其中 t 是尺度参数，$t \in \mathbb{R}$。此处将 $\texttt{normalize}(\boldsymbol{v}_a)$ 中的值视作随机变量 X。RWSig 将尺度参数设为 d 个等差的参数，并将得到的 d 个 $K(t)$ 拼接起来，最终得到 RWSig：

$$\mathrm{RWSig_X} = [\mathrm{K_X(t_i)}]_{\mathrm{t_1},\ldots,\mathrm{t_d}}. \tag{6.21}$$

对于每个 $\texttt{normalize}(\boldsymbol{v}_a)$ 来说，他们都将转换为一个 d 维的向量。

实际上 RWSig 中的每一个值都有它的实际含义。累积量 κ_n 可以通过累积量方程的幂次展开得到：

$$K_X(t) = \sum_{n=1}^{\infty} \kappa_n \frac{t^n}{n!} \tag{6.22}$$

第一个累积量是分布的均值，第二个累积量是方差，第三个累积量是偏度等等。这些值都是直接由 G_i 的结构决定的。比如说，均值与 G_i 中包含的节点个数负相关。方差与 G_i 中节点连接的同质性相关。此处，同质性指的是节点度数分布是否可以由一个良定义的单峰分布刻画。RWSig 中的每个元素实际上是这些累积量的加权求和，位置越靠后，对于高阶累积量的权重越高。

累积量方程有三个优势，与本任务的最终目标十分吻合：

1. 累积量与分布是一一对应的。一个分布对应特定的一组累积量。这也就是说，RWSig 可以作为节点局部结构的唯一标识。
2. $K(t)$ 只与 $\texttt{normalize}(\boldsymbol{v})$ 中的值相关，而与顺序无关。这大大方便了 RWSig 的计算，无需将两个待比对的节点的局部结构对齐，即可计算其相似性。
3. RWSig 仍旧保留了可加性。也就是说相似的网络结构也会有相似的 RWSig。对于不相关的两个变量 X 和 Y 来说

$$K_{X+Y}(t) = \log \mathbb{E}[e^{t(X+Y)}] = \log \mathbb{E}[e^{tX}] + \log \mathbb{E}[e^{tY}]. \tag{6.23}$$

公式6.23表明添加或删除一些边不会对 RWSig 带来巨大的变化。由于原来的向量 $\boldsymbol{v}$ 和结构稍微变化后的新向量 $\boldsymbol{v}'$ 十分接近，二者之差 $\boldsymbol{v} - \boldsymbol{v}'$ 接近一个 0 向量，这使得 $K_{\boldsymbol{v}-\boldsymbol{v}'}(t)$ 也十分接近。RWSig_{v} 和 $\text{RWSig}_{\text{v}'}$ 因此也是相似的。同样的证明也使用于以 m 为中心的子图和以 n 为中心的子图十分相似的情况。RWSig 与值的顺序无关，$\boldsymbol{v}_m$ 和 $\boldsymbol{v}_n$ 由相似的值组成，他们所对应的 $\text{RWSig}_{\text{v}_\text{m}}$ 和 $\text{RWSig}_{\text{v}_\text{n}}$ 也是很相似的。

用随机游走和累积量方程的技术刻画局部结构一方面降低了算法的复杂度，另一方面也给计算带来了一定的随机扰动，这为之后的应用带来一定的隐患。因此 RWSig 设计了两条应对随机扰动的措施。

引入虚数单元 i。RWSig 在公式6.20中引入虚数单元，使得 RWSig 在任意实数 t 上都是良定义。$e^{\mathrm{i}tx} = \cos tx + \mathrm{i}\sin tx$ 的实数和虚数部分都有确定的上下界。

$$K_X(t) = \log \mathbb{E}[e^{\mathrm{i}tX}]. \tag{6.24}$$

公式6.24也叫做第二特征方程[181]。基于公式6.24的新 RWSig 写作：

$$\text{RWSig} = [\text{Real}(\text{K}(\text{t}_\text{i})), \text{Imag}(\text{K}(\text{t}_\text{i}))]_{\text{i}=1,\ldots,\text{d}}. \tag{6.25}$$

保留前 k 个常访问的节点 $\boldsymbol{v}$ 包含 n 个数值，由于图的稀疏性，其中的大部分值都为 0。公式6.24的计算只涉及到非零的成员。RWSig 进一步对非零的成员做了限制，将值过小的成员舍去，而只保留前 k 个经常访问的节点。

RWSig 的实现如伪代码所示。注意，第24和第25计算的是复数 m 取对数后的实数

和虚数部分。

Algorithm 3 Random Walk Based Structural Signature

```
// Initialize the visiting vector
v[0, 1, ..., n − 1] ← 0
// Produce a collection of w-steps-long random walks started from node i and record the visiting times by v[]
for i = 0; i < p; i++ do
    s ← i
    for j = 0; j < w; j++ do
        s ← RandomNeighbor(s)
        v[s] ← v[s] + 1
    end for
end for
// Normalization as Eq.6.19
v ← normalize(v)
// Preserve the frequent visiting nodes
v ← sort(v)
for i = 0; i < opt · n; i++ do
    tmp[i] ← exp(complex(0, v[i]))
end for
// Calculate the cumulants by Eq.6.24 and produce RWSig by Eq.6.25.
for i = 0; i < length(t); i++ do
    for j = 0; j < length(tmp); i++ do
        vec[j] ← pow(tmp[j], t[i])
    end for
    m ← mean(vec)
    sig[2 · i] = log(sqrt(pow(m.real, 2) + pow(m.img, 2)))
    sig[2 · i + 1] = arctan(m.img/m.real)
end for
output sig
```

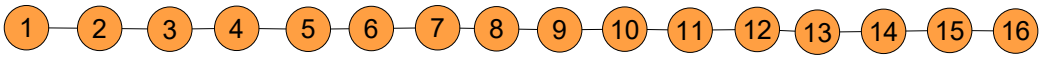
6.2.3　RWSig 与图谱的关系

考虑一个无向图 G，包含 n 个节点和 m 条边。邻接矩阵 $\boldsymbol{A}$ 是对称的，其中为 1 的元素代表所对应的两个节点在图中相连，反之为 0。度向量 $\boldsymbol{d} = \boldsymbol{A}\mathbf{1}$，其中 $\mathbf{1}$ 是一个维度为 $n \times 1$ 的全 1 向量。$\boldsymbol{D}$ 是一个以度数为对角线元素的矩阵：$\boldsymbol{D} = \mathtt{diag}(\boldsymbol{d})$。$\mathcal{L}$ 是图 G 的标准的拉普拉斯矩阵。

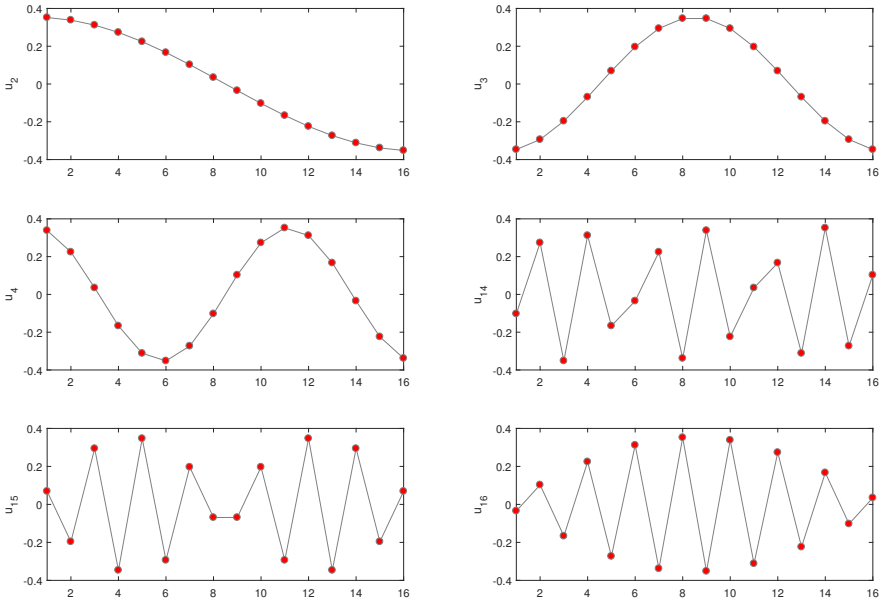
$$\mathcal{L} := \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{L}\boldsymbol{D}^{-\frac{1}{2}} = \boldsymbol{D}^{-\frac{1}{2}}(\boldsymbol{D}-\boldsymbol{A})\boldsymbol{D}^{-\frac{1}{2}} = \boldsymbol{I}-\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}. \tag{6.26}$$

谱分解理论保证了存在一个正交矩阵 $\boldsymbol{U}$ 来对角化 $\mathcal{L}$，

$$\mathcal{L} = \boldsymbol{U}\Lambda\boldsymbol{U}^{\mathrm{T}}, \tag{6.27}$$



(a) 一个由 16 个节点组成的链状图。



(b) 六个特征向量的可视化效果。

图 6.12　一张链状图的拉普拉斯矩阵特征向量可视化。图中绘制第 2nd, 3rd, 4th,14th, 15th and 16th 个拉普拉斯矩阵特征向量。

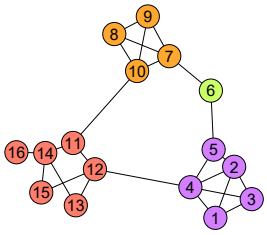
其中 Λ 是一个由非负的特征值做为对角线元素的对角矩阵。$\boldsymbol{U}$ 的每一列是一个特征向量：$\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_n$，特征向量与特征值 $\lambda_1 \leq \lambda_2 \leq \lambda_3 \leq ... \leq \lambda_n$ 一一对应. 尤其是特征向量 $\boldsymbol{u}_1 = \mathbf{1}$ 且对应的特征值 $\lambda_1 = 0$.

图信号的滤波器是一种作用在特征空间的算子，作用的具体形式与特征值相关。

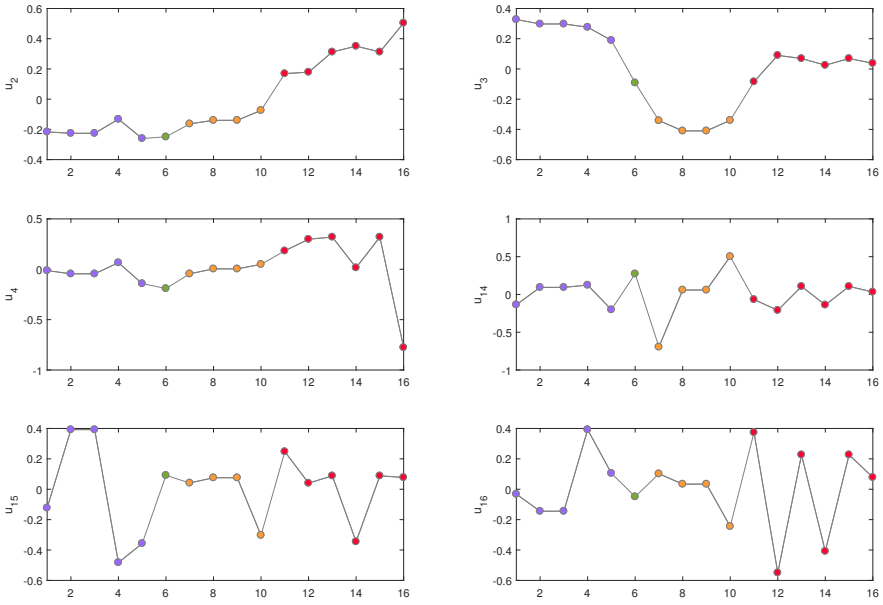
$$\boldsymbol{H} = \boldsymbol{U} h(\Lambda) \boldsymbol{U}^{\mathrm{T}}. \tag{6.28}$$

滤波器 $\boldsymbol{H}$ 是由 $h(x)$ 决定的。

特征向量可以直观地被解释为不同频率的图信号傅立叶变换基。在图6.12和6.13中可视化了拉普拉斯矩阵的特征向量，从而对这一类比有更直接的理解。图6.12是一个简单的链状图，包含 16 个节点。特征向量每个位置的值被展示在图6.12b中，包括有代表性的 $\boldsymbol{u}_2, \boldsymbol{u}_3, \boldsymbol{u}_4, \boldsymbol{u}_{14}, \boldsymbol{u}_{16}$。从图像效果来看，特征向量与传统的正弦波、余弦波十分相似。特征值的大小与波的频率有对应的关系。在图6.13a中，相同颜色的节点在图中相互连结更加紧密。特征向量的每一维与节点编号一一对应，相对应的特征向量元素标记为与节点相一致的颜色。可以发现，当特征值较小时，特征向量的图像看起来更平缓。标记为相同颜色的元素有着很相近的值。相反，当特征值大时，特征向量每个位置的值波动幅度很大。



(a) 一个由 16 个节点组成的，包含社区结构的图。



(b) 六个特征向量的可视化效果。

图 6.13　一张普通图的拉普拉斯矩阵特征向量可视化。图中绘制第 2nd, 3rd, 4th,14th, 15th and 16th 个拉普拉斯矩阵特征向量。

如果类比信号处理中的滤波器，同样可以为图信号设计相似功能的滤波器。在上文的论述中已经展示了特征向量与基波，特征向量与频率之间的类比关系。可以想见，一个图信号的低通滤波器可以是一个与特征值相关的函数。举例来说，图神经网路 GCN[76] 中以 $h(\lambda_i) = 2 - \lambda_i$ 为滤波器，将高频信号抹去。

基于随机游走的 RWSig 与拉普拉斯矩阵的特征向量有着一定的关系。考虑一个有向图 G 上的随机游走过程（无向图中的边可以被视作两条单向边），每个节点在选择下一步的跳转节点时等概率的选择一个邻居节点。令 $\boldsymbol{D}_{\text{out}} = \text{diag}(\sum_{\text{j}} \text{A}_{\text{ij}})$，$\boldsymbol{P}$ 为随机游走的转移矩阵：

$$\boldsymbol{P} = \boldsymbol{D}_{\text{out}}^{-1}\boldsymbol{A} = \boldsymbol{D}_{\text{out}}^{-\frac{1}{2}}(\boldsymbol{I} - \mathcal{L})\boldsymbol{D}_{\text{out}}^{\frac{1}{2}} \sim \boldsymbol{I} - \mathcal{L}. \tag{6.29}$$

$\boldsymbol{P}$ 与 $\boldsymbol{I} - \mathcal{L}$ 相似，因此两个矩阵的特征值是一致的。

$$\lambda_i(\boldsymbol{P}) = 1 - \lambda_i(\mathcal{L}), \tag{6.30}$$

其中 $\lambda_i(\mathcal{L})$ 是拉普拉斯矩阵的第 i^{th} 个特征值。拉普拉斯矩阵最小的特征值为 0，相对

应的 $\boldsymbol{P}$ 的特征值 $\lambda_0(\boldsymbol{P})$ 等于 1，特征向量 $\boldsymbol{u}_0(\boldsymbol{P})$ 通常被称为随机游走特征向量中心度 *random walk eigenvector centrality*。注意，这与一般的邻接矩阵特征值中心度并不是同一个指标。根据随机游走特征向量中心度的定义，$\boldsymbol{u}_0(\boldsymbol{P})$ 中第 i^{th} 个元素代表了随机游走在节点 i 上花费的时间占比。

图神经网络利用低频特征向量去刻画网络拓扑结构和节点特征之间的相关性。RWSig 通过蒙特卡罗方法近似地得到了每个节点的时间占比，间接地利用低频特征向量来刻画每个节点所处的局部结构。实际上，相比特征向量，RWSig 可以更加直观地建模结构。在随机游走中，访问一个节点的概率和与之相邻接的节点相关。换句话说，访问概率不仅与每个节点自身的度数相关，也与它所处的位置，邻居相关。RWSig 考虑局部所涉及节点访问概率的分布，能够以编码的形式为局部结构的刻画提供丰富的信息。

除了可解释性之外，RWSig 与特征分解的方法相比也更加高效鲁棒。特征分解本身是一个困难的计算，尤其对于大规模的图而言，是一个计算的瓶颈。RWSig 不会对几条边的增减敏感，这种轻微的结构变化却会对特征分解的结果造成很大的变化。

6.2.4 实验测评

本节首先验证有关不同网络结构表示算法鲁棒性的论断。其次，会展示多个算法在节点划分和跨结构洞节点识别的问题上的表现。

6.2.4.1 对比算法及参数设定

实验选取了两个构造节点解构特征的方法 (struc2vec[182] 和 GraphWave[179])；两个致力于刻画节点所处局部结构的图表示学习算法 (DeepWalk[10] and node2vec[66])。

1. struc2vec (**s2v**): 该方法通过邻域节点的度数序列来捕捉节点所处的局部结构，并通过构造一个多层的节点关系图来捕捉节点结构的相似性。在本实验中，原始论文中提及的优化方式全被采纳。关系图的层数为 6。
2. GraphWave (**gw**) : 该方法利用图信号的小波变换来刻画节点所处结构。原论文中提出自动计算小波尺度参数的方法，本实验采用了这一做法。
3. DeepWalk (**dw**): 该方法利用随机游走和经典的 word2vec 模型来得到每个节点的向量表示。本实验中采用了与原论文一样的参数设置[10]。
4. node2vec (**n2v**): 该方法声称可以通过平衡深度优先和宽度优先游走，以达到捕捉节点结构的目标。实验中，每个数据集上都手动调整平衡的参数，使得结果最优。DeepWalk 是 node2vec 的一个特例，因此在下面的实验中，有时 node2vec 与 DeepWalk 的结果是一致的。

5. RWSig (**sig**): 本章提出的基于随机游走的算法。窗口大小被设置为 4。以每个节点为起点产生 1000 条游走路径。尺度参数设为 $[1, 2, ..., 100]$，也就是说在公式6.25中，$d = 100$。参数 opt 被设为 30%，这意味着在计算 RWSig 时，只有最常访问的前 30% 节点参与到计算之中。
6. concatenation (**dw⊕sig**): 将 RWSig 与 Deepwalk 的表示向量拼接在一起。

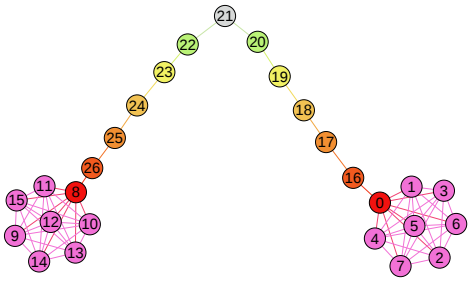
6.2.4.2　鲁棒性分析

如图6.14a所示，实验手动地构造了“杠铃”网络，用一条长的节点链连接两个完全子图。节点的颜色代表了每个节点的结构特征，该颜色在图6.14a到6.14d，图6.14e到6.14h都是一一对应的，方便在多张图中跟踪相同的节点。在本实验中，所有方法产生的表示向量都固定为 16 维，并使用 PCA 降维的方法将其降至 2 维，以此为 x-y 坐标位置绘制节点。

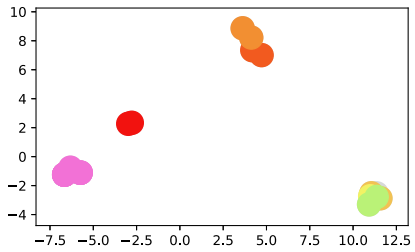
GraphWave(图6.14d) 可以提供最精确的节点结构特征表示，每个相同颜色的节点都在二维向量空间中完全重合。注意，在此处展示的结果与 GraphWave 原论文[179] 中的并不一致。在本实验中使用原文中提供的自动学习尺度参数 s 的方法，而原论文实验是通过手动调整出的参数。在自动学习的模式下，有一些颜色不同的节点是不能够分开的，比如一系列黄色的节点。struc2vec(图6.14c) 和 RWSig (图6.14b) 由于都基于采样和近似计算，不能够完全准确地将同一结构特征的节点投影到同一位置。不过，总的来说，同一颜色的节点在向量空间中的位置是接近的。RWSig 进一步地保持了相似结构节点之间的关系，有相似颜色的节点在向量空间中距离也较为接近。这种关系在 struc2vec 几乎没被保留下来。比如，红色节点（0，8）与橘红色节点（16，17，25，26）相距甚远。

实验中试着删除其中一个完全子图的一条边（边 [1,4]），来观察每个算法对这一变化的敏感度。一个理想的变化方向是：在向量空间中，节点 1、节点 4 与原完全子图中的其他节点之间有一定的间隔，不过仍是聚集在一起的，节点 1、节点 4 与其他节点之间距离较大。struc2vec 完全不能识别出节点 1、节点 4 与原完全子图中的其他节点之间的间隔，该算法对一条边的缺失太不敏感。GraphWave 将节点 1、节点 4 与原完全子图中的其他节点完全分离，但该算法过于敏感，这些节点在表示空间中并不聚集。从这两个角度来看，RWSig 是一个相对理想的刻画节点结构的工具。

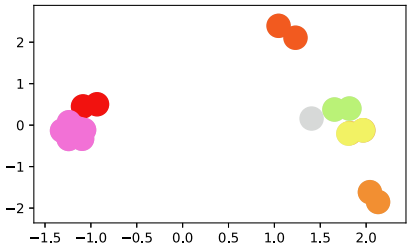
在表6.2中，测试了不同算法在面对不同程度上的结构变化时的表现。以某个用户为中心，抽出了其在 Facebook 中社交网络，包含多跳邻居节点。该网络 G 包含 334 个节点，2852 条边。在实验中，随机保留图中 p% 的边，得到网络 G'。计算 G 和 G' 中对应节点之间结构表示向量之间的差距。由于这些表示向量通常被用作下游任务模型的输入特征，当出现微小的结构变化时，理想的表示向量变化应是相对平滑的。实



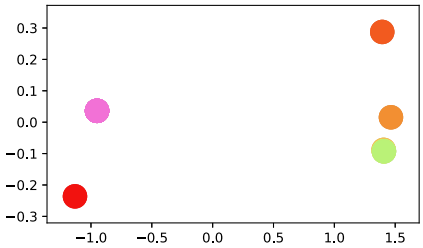
(a) 杠铃网络。



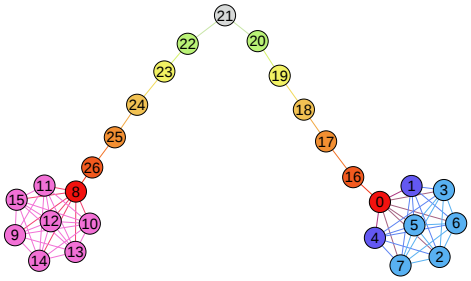
(b) RWSig.



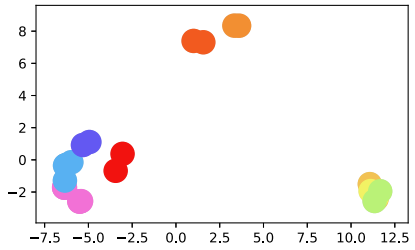
(c) struc2vec.



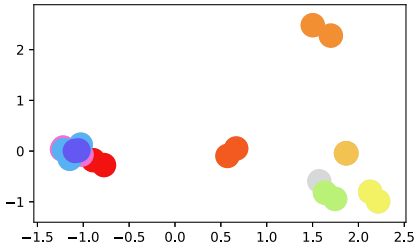
(d) GraphWave.



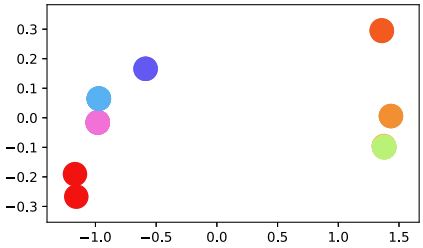
(e) Delete the edge (1,4).



(f) RWSig.



(g) struc2vec.



(h) GraphWave.

图 6.14　利用几个节点结构向量表示学习方法来处理 (a) 杠铃网络和 (e) 缺少了 (1,4) 边的杠铃网络。利用 PCA 降维[183]的方法，学习到的节点表示向量被影射到二维空间。

验计算了 G 和 G' 中对应节点之间结构表示向量之间的平均欧式距离和平均 cosine 距离，以观察不同算法给出的结果的变化程度。如表6.2所示，随着移除边的概率的上升，RWSig 和 GraphWave 的欧式距离在逐步增大，cosine 相似度在逐步下降。在所有测试的算法中，RWSig 结果的变化相对最小。即便网络结构未发生任何改变，DeepWalk 也不能够给出一个稳定不变的结果。这是因为 DeepWalk 的训练依赖于随机梯度下降。也

表 6.2　图中边保持不变的比例变化时，节点结构特征向量的变化。

指标	方法	95%	90%	85%	80%	75%	70%	65%	60%
Euclidean Distance	sig	0.297	0.362	0.432	0.521	0.658	0.828	0.963	1.289
	s2v	2.770	2.747	2.845	3.129	3.065	3.122	3.262	3.358
	gw	0.649	0.693	0.891	1.204	1.500	1.731	2.060	2.230
	dw	3.697	3.780	3.747	3.765	3.813	3.821	3.854	3.905
Cosine Similarity	sig	0.996	0.993	0.991	0.985	0.977	0.973	0.967	0.953
	s2v	0.208	0.232	0.190	0.140	0.165	0.131	0.123	0.107
	gw	0.990	0.989	0.984	0.969	0.961	0.957	0.946	0.940
	dw	0.017	-0.017	0.008	0.013	0.008	0.002	0.008	-0.009

表 6.3　跨结构洞节点的识别。表中展示的是训练比例为 1% 到 9% 时的 micro 和 macro F1 值。

方法	指标	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
dw+sig	macro	**0.44**	**0.45**	**0.45**	**0.45**	**0.46**	**0.46**	**0.46**	**0.46**	**0.46**
	micro	**0.45**	**0.45**	**0.46**	**0.46**	**0.46**	**0.46**	**0.46**	**0.46**	**0.46**
sig	maco	0.3	0.31	0.31	0.31	0.31	0.31	0.31	0.31	0.31
	micro	0.39	0.4	0.4	0.39	0.39	0.4	0.4	0.39	0.39
dw	macro	0.36	0.37	0.37	0.37	0.37	0.37	0.38	0.38	0.38
	micro	0.37	0.37	0.38	0.38	0.38	0.38	0.38	0.38	0.38

正是因为这个原因，可以发现，当 $p = 95\%$ 时，DeepWalk 的 consine 相似度仍然接近 0。struc2vec 和 DeepWalk 涉及到同样的优化步骤，因此也会出现结果完全不相关，波动巨大的状况。$p = 90\%$ 和 $p = 95\%$ 相比，struc2vec 有着更小的欧式距离和更大的 cosine 相似度。总而言之，这个实验验证了 RWSig 随着网络结构的变化而稳定地变化，而且对于微小的结构变化，方法是鲁棒的。

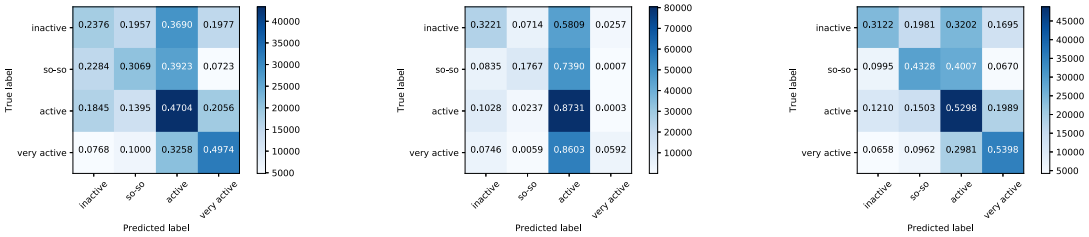
6.2.4.3　跨结构洞节点的识别

本实验利用 RWSig 来识别图中的跨结构洞节点，这些节点在图中往往横跨多个社区。实验在 DBLP 的作者合作网络上进行。在图中，节点是作者，边代表两个作者共同署名发表了文章，社区为发表的会议。网络包含 317K 的节点和 1M 的边，3K 的社区。对结构进行编码的图表示学习算法都无法处理如此庞大的网络。因此在这个实验中，对比算法只有 DeepWalk。

DBLP 数据集本身包含了节点所属的社区，实验用节点所属的社区数目和节点度数之比来衡量社区的活跃度，该定义与[184]一致。

$$a_i = \frac{\#(community_i)}{deg(i)}. \tag{6.31}$$

实验中，将所有节点的活跃度按照四分位分为四个活跃等级。用一部分数据训练 logistic 分类器，以判断节点的活跃等级。表6.3 展示了训练比例为 1% 到 9% 时分类器的分类



(a) DeepWalk. (b) RWSig. (c) Concatenation.

图 6.15 通过可视化混淆矩阵，进一步分析分类器在各个类别上的表现差异。

表现。拼接的向量相比任何两个未拼接的向量都有很大的提升。本文尝试进一步挖掘提升的根源从何而来。因此在此将分类的混合矩阵绘制出来，如图6.15所示。单纯的依赖局部的结构信息不是个好主意。尤其在 DBLP 这个数据中，图6.15b说明 RWSig 倾向于将更多的节点分为活跃的等级。然而对于非活跃的节点，RWSig 有着更好的表现。图6.15a说明 DeepWalk 对于活跃和非活跃的节点有着比较均衡的表现。拼接后的向量在三个类别上都有更好的表现。不活跃的节点虽然效果仍然不是很好，但也比 DeepWalk 更好了，这要归功于 RWSig。总的来说，相比于活跃的节点，对于不活跃节点的探测要更多的依赖节点的结构信息。活跃的研究者通常在合作图中与其他的活跃研究者有着紧密的连接，然而不活跃的研究者通常只有少量的边，而且与他相连的节点有活跃也会有不活跃的。

6.3 本章小结

本章重点关注路径解构和邻域解构策略相结合的问题。首先提出考虑节点所处局部结构相似性的图表示学习算法 — SNS。SNS 将两个节点所处的导出子图相似度计算分解为两个节点参与特殊的小子图个数对比的子问题。每个子问题仅需关注目标节点和其邻居节点构成的特殊子图（如三角形、K4 完全图等）。SNS 将两个节点在图上连接的紧密度计算分解为判断两个节点在同一随机游走窗口内是否共同出现的子问题。每个子问题只需关注一个相关的随机游走路径窗口。在本章中分析了基于随机游走采样的传统图表示学习算法的劣势，并通过一系列的实验证明其无法刻画节点的结构相似性。同时在计算导出子图的相似性这一任务上，进一步分析了不同特殊小子图的重要程度，从而进一步提升算法的效果。该工作已经发表在 CIKM2017 上。

随后，本文继续关注节点所处局部结构的表示方法这一问题，希望应用其于跨结构洞节点的识别。在 SNS 算法中，利用邻域解构的策略来刻画导出子图的相似度计算

复杂度较高，不适用于大规模的图。本章提出的 RWSig 方法试图利用随机游走访问节点次数的分布来解决这一问题。从设计思想上讲，该方法将节点所处的导出子图分解为多个特殊的子图 — 随机游走路径，可以看作是路径解构和邻域解构策略的融合。这一方法不仅极大的提升了刻画节点所处结构的速度和准确性，也为识别跨结构洞节点提供了新的可能性。该工作目前在审稿中。

第七章　总结与展望

7.1　本文工作及创新性

根据维基百科定义，图 (graph) 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)。各式各样的图结构在日常生活中非常普遍。

随着互联网的发展，大规模的线上购物、社交平台等等不断涌现。这些平台吸引了海量的用户，同时获取了大量各种形式的图数据。与传统的图相比，利用这些大规模图有以下三个挑战：

1. **临近度计算复杂。**图挖掘通常围绕着图中节点之间的关系进行计算。最短路径、最大流 — 最小割问题、节点中心度度量都是传统图挖掘中经典的问题。然而当图的规模不断增长，这些问题的算法复杂度成为了其主要的限制因素。大规模图要求更高效、快捷的图挖掘算法。
2. **图结构信息缺失。**由于数据采集方式的不足，图数据通常都是不完整的。边、点、以及各类属性信息都有可能缺失。同时有很多结构信息不可被显式地捕捉，但是可以隐式地推测。这要求图挖掘算法有更强的鲁棒性和泛化能力。
3. **同质性形式多样。**同质性指的是“图中，相似的个体 (节点) 相互连结的概率”。信息的来源是多样的，这使得图包含多种节点类型、边类型，节点之间的相似性亦可以从多个维度来定义。对于图挖掘算法而言，厘清图中的连边和任务所关注的某几个维度之间的关系变得重要而困难。

针对这三大挑战，本文总结已有的图挖掘经典算法中与分治思想相似的设计方法，并将其称为图的解构策略。简单来说，图的解构策略是将原始图挖掘任务拆解为多个子问题，求解每个子问题时只需围绕原图的一个子图进行计算，原问题可以在聚合各个子问题结果的基础之上得到解决。解构在工程中是一个常规的策略，譬如分治算法和模块化的设计方式，都是将一个任务分解为多个小任务，并分别处理多个小任务。具体到图挖掘的任务，图的解构策略赋予算法三种能力：

1. **细化邻接关系的能力。**审视节点邻接关系的角度随着解构方式的不同而变化，因此算法可以从更多的角度刻画节点之间的关系。
2. **组合的泛化能力。**算法的子问题求解对应图的解构，而聚合各个子问题的结果则对应图的重构。解构和重构是增强算法泛化能力的常规操作，在一定程度上缓解了图结构不完整的问题。
3. **化繁为简的能力。**图中复杂度较高的运算被以随机游走为代表的简便运算替代，

大大加快了算法的运算速度，增强了算法处理大规模图的能力。

本文根据划分出的子图结构，将图的解构分为路径解构和邻域解构，并详细阐释了两种解构方式在图挖掘中的应用实例。具体包括:

1. **邻域解构的运用**。本文提出一个体现邻域解构的大规模社区发现算法，并将其应用于节点传播影响力的计算中。算法将社区发现任务分解为判断一对邻接节点是否同属于一个社区的子问题，每个子问题只围绕目标节点的自我网络进行计算。
2. **路径解构的运用**。本文提出一个体现路径解构的图神经网络，并通过分析其背后的数理含义，提出一个新的节点中心度指标。算法将刻画图中一对节点的关系分解为计算随机游走路径中，两个节点依次出现的概率的子问题，每个子问题只围绕一条随机游走路径进行计算。
3. **邻域解构和路径解构的综合运用**。本文提出一个利用两种解构方式分别刻画两种节点相似性的图表示学习算法，并进一步融合两种解构方式，提出一个快速刻画节点地位的方法，实现跨结构洞节点的识别。其中在图表示学习算法中，算法将刻画节点所处局部结构分解为统计特殊小子图的子问题，每个子问题只围绕一种类型的特殊小子图进行计算。为了进一步加速这一过程，在应用中，本文使用随机游走路径替代特殊小子图，对目标节点的邻域进行路径解构，实现了邻域解构和路径解构的结合。

7.2 未来工作展望

围绕图的解构策略在图挖掘中的应用，本文的研究工作还有很多可以延续和深入扩展的地方，在此列举几个重要方面。

- 图的解构策略
    1. 除路径解构和邻域解构以外，还有哪些有效解构形式？本文在总结已有工作的基础之上，提炼出图的解构策略，并将其划分为两种解构形式。而这并不代表不存在其他形式的解构，或者更加恰当的划分方式。尤其是当图的形式更加复杂时，异构图、时序图、知识图谱等等是否存在其他类型的解构方式？深入分析这一问题，可以为新算法的设计带来更多的启发。
    2. 能否实现自动解构？本文所提及的所有算法及应用都是手动设置解构的操作，解构的子图是有限的，固定的。在机器学习技术铺天盖地的未来，能否通过算法自动地学习出更有意义，更符合应用背景的子图结构？自动化可以为图的解构技术带来更多的灵活性，激活更大潜力。

3. 多领域中的图的解策略。本文仅仅局限于图挖掘算法与应用中的图的解构策略。实际上，在工业界，一个上线的图计算任务还涉及到图数据库的分布式存储，图的高性能分布式计算等。探索图的解构策略与相关领域的联系是必要的。

- 图表示算法与应用
  1. 社区发现算法与邻域解构的关系。本文提出的社区发现算法完美体现了邻域解构。事实上，社区发现任务与邻域解构有共通之处，都要将图有目的地划分为多个子图。因此，能否借由社区发现算法的框架设计邻域自动解构的方法呢？这两个子领域的交叉有可能激发新的火花。
  2. 在图表示学习中，如何自动平衡两种解构方式？本文提出的图表示算法利用路径解构刻画节点在连结紧密程度，利用邻域解构刻画节点的地位相似性。两种类型的节点相似度被手动设定规则结合在一起。是否存在更自然的融合方式呢？以及除了这两种节点相似度以外，是否存在其他形式的节点相似度？
  3. 复杂的路径解构与节点中心度的关系。本文仅仅探讨了原始的路径解构与节点中心度之间的关系。然而，随着图表示学习的发展，已经出现了多种图类型、多种形式的路径解构。这些复杂的路径解构与节点中心度之间有什么关系呢？是否可以继续定义一系列的节点中心度指标？
  4. 基于路径解构的统一节点表示框架。本文最后对节点的邻域进行路径解构，以实现对节点地位的向量表示。那么是否可以完全基于路径解构，既能对节点之间的关系进行建模，又可以对节点地位进行建模？倘若成型，那么大规模图的节点向量表示的计算效率又可以大幅度提升。

参考文献

[1] P. Goyal and E. Ferrara. "*Graph embedding techniques, applications, and performance: A survey*". *Knowledge-Based Systems*, **2018**, *151*: 78–94.

[2] S. Bhagat, G. Cormode and S. Muthukrishnan. "*Node classification in social networks*". In: *Social network data analytics*. Springer, **2011**: 115–148.

[3] D. Liben-Nowell and J. Kleinberg. "*The link-prediction problem for social networks*". *Journal of the American society for information science and technology*, **2007**, *58*(7): 1019–1031.

[4] C. H. Ding, X. He, H. Zha *et al.* "*A min-max cut algorithm for graph partitioning and data clustering*". In: *Proceedings 2001 IEEE international conference on data mining*. **2001**: 107–114.

[5] 北岛. 北岛诗选. 新世纪出版社, **1986**.

[6] F. Schweitzer, G. Fagiolo, D. Sornette *et al.* "*Economic networks: The new challenges*". *science*, **2009**, *325*(5939): 422–425.

[7] M. Heideman, D. Johnson and C. Burrus. "*Gauss and the history of the fast Fourier transform*". *IEEE ASSP Magazine*, **1984**, *1*(4): 14–21.

[8] D. E. Knuth. *The art of computer programming*. Pearson Education, **1997**.

[9] X. Xu, L. Lu, P. He *et al.* "*Protein classification using random walk on graph*". In: *International Conference on Intelligent Computing*. **2012**: 180–184.

[10] B. Perozzi, R. Al-Rfou and S. Skiena. "*Deepwalk: Online learning of social representations*". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. **2014**: 701–710.

[11] G. Kossinets and D. J. Watts. "*Empirical analysis of an evolving social network*". *science*, **2006**, *311*(5757): 88–90.

[12] T. Lyu, F. Sun, P. Jiang *et al.* "*Compositional network embedding for link prediction*". In: *Proceedings of the 13th ACM Conference on Recommender Systems*. **2019**: 388–392.

[13] V. Nicosia, G. Mangioni, V. Carchiolo *et al.* "*Extending the definition of modularity to directed graphs with overlapping communities*". *Journal of Statistical Mechanics: Theory and Experiment*, **2009**, *2009*(03): P03024.

[14] J. Yang, J. McAuley and J. Leskovec. "*Community detection in networks with node attributes*". In: *2013 IEEE 13th International Conference on Data Mining*. **2013**: 1151–1156.

[15] P. D. Hoff, A. E. Raftery and M. S. Handcock. "*Latent space approaches to social network analysis*". *Journal of the american Statistical association*, **2002**, *97*(460): 1090–1098.

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

[17] M. M. Bronstein, J. Bruna, Y. LeCun *et al.* "*Geometric deep learning: going beyond euclidean data*". *IEEE Signal Processing Magazine*, **2017**, *34*(4): 18–42.

[18] M. Nickel, K. Murphy, V. Tresp *et al.* "*A review of relational machine learning for knowledge graphs*". *Proceedings of the IEEE*, **2015**, *104*(1): 11–33.

[19] W. W. Zachary. "*An information flow model for conflict and fission in small groups*". *Journal of anthropological research*, **1977**, *33*(4): 452–473.

[20] D. Lusseau. "*The emergent properties of a dolphin social network*". *Proceedings of the Royal Society of London. Series B: Biological Sciences*, **2003**, *270*(suppl_2): S186–S188.

[21] S. Kelley. *The existence and discovery of overlapping communities in large-scale networks* [phdthesis]. **2009**.

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

[23] A. Clauset, M. E. Newman and C. Moore. "*Finding community structure in very large networks*". *Physical review E*, **2004**, *70*(6): 066111.

[24] A. Medus, G. Acuña and C. O. Dorso. "*Detection of community structures in networks via global optimization*". *Physica A: Statistical Mechanics and its Applications*, **2005**, *358*(2-4): 593–604.

[25] V. D. Blondel, J.-L. Guillaume, R. Lambiotte *et al.* "*Fast unfolding of communities in large networks*". *Journal of statistical mechanics: theory and experiment*, **2008**, *2008*(10): P10008.

[26] A. Lancichinetti and S. Fortunato. "*Community detection algorithms: a comparative analysis*". *Physical review E*, **2009**, *80*(5): 056117.

[27] P. Pons and M. Latapy. "*Computing communities in large networks using random walks*". In: *International symposium on computer and information sciences*. **2005**: 284–293.

[28] M. Rosvall and C. T. Bergstrom. "*Maps of random walks on complex networks reveal community structure*". *Proceedings of the National Academy of Sciences*, **2008**, *105*(4): 1118–1123.

[29] I. Farkas, D. Ábel, G. Palla *et al.* "*Weighted network modules*". *New Journal of Physics*, **2007**, *9*(6): 180.

[30] J. M. Kumpula, M. Kivelä, K. Kaski *et al.* "*Sequential algorithm for fast clique percolation*". *Physical Review E*, **2008**, *78*(2): 026109.

[31] F. Havemann, M. Heinz, A. Struck *et al.* "*Identification of overlapping communities and their hierarchy by locally calculating community-changing resolution levels*". *Journal of Statistical Mechanics: Theory and Experiment*, **2011**, *2011*(01): P01023.

[32] D. Jin, B. Yang, C. Baquero *et al.* "*A markov random walk under constraint for discovering overlapping communities in complex networks*". *Journal of Statistical Mechanics: Theory and Experiment*, **2011**, *2011*(05): P05031.

[33] A. Lancichinetti, F. Radicchi, J. J. Ramasco *et al.* "*Finding statistically significant communities in networks*". *PloS one*, **2011**, *6*(4): e18961.

[34] P. K. Gopalan and D. M. Blei. "*Efficient discovery of overlapping communities in massive networks*". *Proceedings of the National Academy of Science*, **2013**, *110*(36): 14534–14539.

[35] J. Yang and J. Leskovec. "*Overlapping community detection at scale: a nonnegative matrix factorization approach*". In: *Proceedings of the sixth ACM international conference on Web search and data mining*. **2013**: 587–596.

[36] J. Yang, J. McAuley and J. Leskovec. "*Detecting cohesive and 2-mode communities indirected and undirected networks*". In: *Proceedings of the 7th ACM international conference on Web search and data mining*. **2014**: 323–332.

[37] R. Kannan, S. Vempala *et al.* "*Spectral algorithms*". *Foundations and Trends® in Theoretical Computer Science*, **2009**, *4*(3–4): 157–288.

[38] M. Chen, Q. Yang and X. Tang. "*Directed Graph Embedding.*" In: *IJCAI*. **2007**: 2707–2712.

[39] J. Tang, M. Qu, M. Wang *et al.* "*Line: Large-scale information network embedding*". In: *Proceedings of the 24th international conference on world wide web*. **2015**: 1067–1077.

[40] D. Wang, P. Cui and W. Zhu. "*Structural deep network embedding*". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. **2016**: 1225–1234.

[41] T. Martin. "*community2vec: Vector representations of online communities encode semantic relationships*". In: *Proceedings of the Second Workshop on NLP and Computational Social Science*. **2017**: 27–31.

[42] T. Lou, J. Tang, J. Hopcroft *et al.* "*Learning to predict reciprocity and triadic closure in social networks*". *ACM Transactions on Knowledge Discovery from Data (TKDD)*, **2013**, *7*(2): 5.

[43] D. Cartwright and F. Harary. "*Structural balance: a generalization of Heider's theory.*" *Psychological review*, **1956**, *63*(5): 277.

[44] M. Ou, P. Cui, J. Pei *et al.* "*Asymmetric transitivity preserving graph embedding*". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. **2016**: 1105–1114.

[45] C. Tu, W. Zhang, Z. Liu *et al.* "*Max-margin deepwalk: Discriminative learning of network representation.*" In: *IJCAI*. **2016**: 3889–3895.

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

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

[48] C. Yang, Z. Liu, D. Zhao *et al.* "*Network representation learning with rich text information*". In: *Twenty-Fourth International Joint Conference on Artificial Intelligence*. **2015**.

[49] S. Yeung, O. Russakovsky, G. Mori *et al.* "*End-to-end learning of action detection from frame glimpses in videos*". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. **2016**: 2678–2687.

[50] X. Yang, Y.-N. Chen, D. Hakkani-Tür *et al.* "*End-to-end joint learning of natural language understanding and dialogue manager*". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. **2017**: 5690–5694.

[51] C. Li, J. Ma, X. Guo *et al.* "*Deepcas: An end-to-end predictor of information cascades*". In: *Proceedings of the 26th international conference on World Wide Web*. **2017**: 577–586.

[52] R. Hu, C. C. Aggarwal, S. Ma *et al.* "*An embedding approach to anomaly detection*". In: *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. **2016**: 385–396.

[53] T. Man, H. Shen, S. Liu *et al.* "*Predict Anchor Links across Social Networks via an Embedding Approach.*" In: *IJCAI*. **2016**: 1823–1829.

[54] T. Chen and Y. Sun. "*Task-guided and path-augmented heterogeneous network embedding for author identification*". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. **2017**: 295–304.

[55] P. W. Battaglia, J. B. Hamrick, V. Bapst *et al.* "*Relational inductive biases, deep learning, and graph networks*". *arXiv preprint arXiv:1806.01261*, **2018**.

[56] P. Veličković, G. Cucurull, A. Casanova *et al.* "*Graph attention networks*". *arXiv preprint arXiv:1710.10903*, **2017**.

[57] W. Hamilton, Z. Ying and J. Leskovec. "*Inductive representation learning on large graphs*". In: *Advances in Neural Information Processing Systems*. **2017**: 1024–1034.

[58] M. Defferrard, X. Bresson and P. Vandergheynst. "*Convolutional neural networks on graphs with fast localized spectral filtering*". In: *Advances in neural information processing systems*. **2016**: 3844–3852.

[59] A. Kusiak. "*Decomposition in data mining: An industrial case study*". *IEEE transactions on electronics packaging manufacturing*, **2000**, *23*(4): 345–353.

[60] W. L. Buntine. *Graphical Models for Discivering Knowledge.* **1996**.

[61] M. J. Zaki and C.-T. Ho. *Large-scale parallel data mining*. Springer Science & Business Media, **2000**.

[62] C. Zhang and Y. Ma. *Ensemble machine learning: methods and applications*. Springer, **2012**.

[63] D. J. Miller and H. S. Uyar. "*A mixture of experts classifier with learning based on both labelled and unlabelled data*". In: *Advances in neural information processing systems*. **1997**: 571–577.

[64] A. J. C. SHARKEY. "*On combining artificial neural nets*". *Connection Science*, **1996**, *8*(3-4): 299–314.

[65] L. Y. Pratt, J. Mostow, C. A. Kamm *et al.* "*Direct Transfer of Learned Information Among Neural Networks.*" In: *AAAI*. **1991**: 584–589.

[66] A. Grover and J. Leskovec. "*node2vec: Scalable feature learning for networks*". In: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. **2016**: 855–864.

[67] H. Chen, B. Perozzi, Y. Hu *et al.* "*Harp: Hierarchical representation learning for networks*". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. **2018**.

[68] B. Perozzi, V. Kulkarni and S. Skiena. "*Walklets: Multiscale graph embeddings for interpretable network classification*". *arXiv preprint arXiv:1605.02115*, **2016**.

[69] B. P. Chamberlain, J. Clough and M. P. Deisenroth. "*Neural embeddings of graphs in hyperbolic space*". *arXiv preprint arXiv:1705.10359*, **2017**.

[70] S. Cao, W. Lu and Q. Xu. "*Deep neural networks for learning graph representations*". In: *Thirtieth AAAI Conference on Artificial Intelligence*. **2016**.

[71] S. P. Borgatti, A. Mehra, D. J. Brass *et al.* "*Network analysis in the social sciences*". *science*, **2009**, *323*(5916): 892–895.

[72] M. Everett and S. P. Borgatti. "*Ego network betweenness*". *Social networks*, **2005**, *27*(1): 31–38.

[73] R. S. Burt. *Structural holes: The social structure of competition*. Harvard university press, **2009**.

[74] T. N. Kipf and M. Welling. "*Variational graph auto-encoders*". *arXiv preprint arXiv:1611.07308*, **2016**.

[75] M. Schlichtkrull, T. N. Kipf, P. Bloem *et al.* "*Modeling relational data with graph convolutional networks*". In: *European Semantic Web Conference*. **2018**: 593–607.

[76] T. N. Kipf and M. Welling. "*Semi-Supervised Classification with Graph Convolutional Networks*". **2017**: 1–14.

[77] T. Pham, T. Tran, D. Phung *et al.* "*Column networks for collective classification*". In: *Thirty-First AAAI Conference on Artificial Intelligence*. **2017**.

[78] T. M. Newcomb. "*The acquaintance process as a prototype of human interaction.*" **1961**.

[79] R. L. Collins. "*For better or worse: The impact of upward social comparison on self-evaluations.*" *Psychological bulletin*, **1996**, *119*(1): 51.

[80] M. E. Newman. "*A measure of betweenness centrality based on random walks*". *Social networks*, **2005**, *27*(1): 39–54.

[81] L. C. Freeman, S. P. Borgatti and D. R. White. "*Centrality in valued graphs: A measure of betweenness based on network flow*". *Social networks*, **1991**, *13*(2): 141–154.

[82] K. Stephenson and M. Zelen. "*Rethinking centrality: Methods and examples*". *Social networks*, **1989**, *11*(1): 1–37.

[83] S. Milgram. "*The small world problem*". *Psychology today*, **1967**, *2*(1): 60–67.

[84] J. Travers and S. Milgram. "*An experimental study of the small world problem*". In: *Social Networks*. Elsevier, **1977**: 179–197.

[85] P. S. Dodds, R. Muhamad and D. J. Watts. "*An experimental study of search in global social networks*". *science*, **2003**, *301*(5634): 827–829.

[86] X. Zhao, A. Chang, A. D. Sarma *et al.* "*On the embeddability of random walk distances*". *Proceedings of the VLDB Endowment*, **2013**, *6*(14): 1690–1701.

[87] T. Lyu, Y. Zhang and Y. Zhang. "*Enhancing the network embedding quality with structural similarity*". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. **2017**: 147–156.

[88] M. McPherson, L. Smith-Lovin and J. M. Cook. "*Birds of a feather: Homophily in social networks*". *Annual review of sociology*, **2001**, *27*(1): 415–444.

[89] S. M. Goodreau, J. A. Kitts and M. Morris. "*Birds of a feather, or friend of a friend? Using exponential random graph models to investigate adolescent social networks*". *Demography*, **2009**, *46*(1): 103–125.

[90] P. V. Marsden. "*Core discussion networks of Americans*". *American sociological review*, **1987**: 122–131.

[91] L. M. Verbrugge. "*The structure of adult friendship choices*". *Social forces*, **1977**, *56*(2): 576–597.

[92] T. L. Huston and G. Levinger. "*Interpersonal attraction and relationships*". *Annual review of psychology*, **1978**, *29*(1): 115–156.

[93] D. B. Kandel. "*Homophily, selection, and socialization in adolescent friendships*". *American journal of Sociology*, **1978**, *84*(2): 427–436.

[94] J. Ma, A. Shojaie and G. Michailidis. "*Network-based pathway enrichment analysis with incomplete network information*". *Bioinformatics*, **2016**, *32*(20): 3165–3174.

[95] X. Huang, J. Li and X. Hu. "*Label informed attributed network embedding*". In: *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. **2017**: 731–739.

[96] S. Chakrabarti. "*Dynamic Personalized Pagerank in Entity-Relation Graphs*". In: *Proceedings of the 16th International Conference on World Wide Web*. Banff, Alberta, Canada, **2007**: 571–580.

[97] G. Jeh and J. Widom. "*Scaling Personalized Web Search*". In: *Proceedings of the 12th International Conference on World Wide Web*. Budapest, Hungary: Association for Computing Machinery, **2003**: 271–279.

[98] D. Fogaras, B. Rácz, K. Csalogány *et al.* "*Towards Scaling Fully Personalized PageRank: Algorithms, Lower Bounds, and Experiments*". *Internet Mathematics*, **2005**, *2*(3): 333–358.

[99] K. Avrachenkov, N. Litvak, D. Nemirovsky *et al.* "*Monte Carlo Methods in PageRank Computation: When One Iteration is Sufficient*". *SIAM J. Numerical Analysis*, **2007**, *45*(2): 890–904.

[100] D. A. Spielman and N. Srivastava. "*Graph Sparsification by Effective Resistances*". *SIAM J. Comput.* **2011**, *40*(6): 1913–1926.

[101] R.-H. Li, L. Qin, J. X. Yu *et al.* "*Influential community search in large networks*". *Proceedings of the VLDB Endowment*, **2015**.

[102] M. Wang, C. Wang, J. X. Yu *et al.* "*Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework*". *Proceedings of the VLDB Endowment*, **2015**.

[103] X. Huang, H. Cheng, L. Qin *et al.* "*Querying k-truss community in large and dynamic graphs*". In: *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. **2014**.

[104] J. Shao, Z. Han, Q. Yang *et al.* "*Community Detection based on Distance Dynamics*". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. **2015**: 1075–1084.

[105] B. Adamcsek, G. Palla, I. J. Farkas *et al.* "*CFinder: locating cliques and overlapping modules in biological networks*". *Bioinformatics*, **2006**, *22*(8): 1021–1023.

[106] J. M. Kumpula, M. Kivelä, K. Kaski *et al.* "*Sequential algorithm for fast clique percolation*". *Phys. Rev. E*, 2008-08, *78*: 026109.

[107] X. Zhang, C. Wang, Y. Su *et al.* "*A fast overlapping community detection algorithm based on weak cliques for large-scale networks*". *IEEE Transactions on Computational Social Systems*, **2017**.

[108] A. Anandkumar, R. Ge, D. Hsu *et al.* "*A tensor approach to learning mixed membership community models*". *Journal of Machine Learning Research*, **2014**, *15*(1): 2239–2312.

[109] J. J. Whang, D. F. Gleich and I. S. Dhillon. "*Overlapping Community Detection Using Neighborhood-Inflated Seed Expansion*". *IEEE Transactions on Knowledge and Data Engineering*, 2016-05, *28*(5): 1272–1284.

[110] E. Le Martelot and C. Hankin. "*Fast multi-scale detection of overlapping communities using local criteria*". *Computing*, 2014-11, *96*(11): 1011–1027.

[111] J. J. Whang, D. F. Gleich and I. S. Dhillon. "*Overlapping community detection using neighborhood-inflated seed expansion*". *IEEE Transactions on Knowledge and Data Engineering*, **2016**, *28*(5): 1272–1284.

[112] S. Cavallari, V. W. Zheng, H. Cai *et al.* "*Learning community embedding with community detection and node embedding on graphs*". In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. **2017**: 377–386.

[113] X. Wang, P. Cui, J. Wang *et al.* "*Community Preserving Network Embedding.*" In: *AAAI*. **2017**: 203–209.

[114] J. Ugander and L. Backstrom. "*Balanced label propagation for partitioning massive graphs*". In: *Proceedings of the sixth ACM international conference on Web search and data mining*. **2013**: 507–516.

[115] C. Shi, Y. Cai, D. Fu *et al.* "*A link clustering based overlapping community detection algorithm*". *Data & Knowledge Engineering*, **2013**, *87*: 394–404.

[116] S. Jabbour, N. Mhadhbi, B. Raddaoui *et al.* "*A SAT-Based Framework for Overlapping Community Detection in Networks*". In: J. Kim, K. Shim, L. Cao *et al.*, eds. *Advances in Knowledge Discovery and Data Mining*. Cham: Springer International Publishing, **2017**: 786–798.

[117] L. Zhang, H. Pan, Y. Su *et al.* "*A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection*". *IEEE Transactions on Cybernetics*, **2017**, *47*(9): 2703–2716.

[118] S. Athey, E. Calvano and S. Jha. "*A theory of community formation and social hierarchy*". *preprint*, **2006**.

[119] J. Xie, S. Kelley and B. K. Szymanski. "*Overlapping community detection in networks: The state-of-the-art and comparative study*". *ACM Computing Surveys*, **2013**, *45*(4): 43.

[120] L. Zhou, P. Yang, K. Lü *et al.* "*A fast approach for detecting overlapping communities in social networks based on game theory*". In: *Data Science*. Springer, **2015**: 62–73.

[121] R. Arava. "*An Efficient homophilic model and Algorithms for Community Detection using Nash Dynamics*". *arXiv preprint arXiv:1506.05659*, **2015**.

[122] M. Crampes and M. Plantié. "*Overlapping Community Detection Optimization and Nash Equilibrium*". *arXiv preprint arXiv:1406.6832*, **2014**.

[123] W. Chen, Z. Liu, X. Sun *et al.* "*A game-theoretic framework to identify overlapping communities in social networks*". *Data Mining and Knowledge Discovery*, **2010**, *21*(2): 224–240.

[124] R. Narayanam and Y. Narahari. "*A game theory inspired, decentralized, local information based algorithm for community detection in social graphs*". In: *Pattern Recognition (ICPR), 2012 21st International Conference on*. 2012-11: 1072–1075.

[125] A. Prat-Pérez, D. Dominguez-Sal, J. M. Brunat *et al.* "*Shaping Communities out of Triangles*". In: *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. Maui, Hawaii, USA: Association for Computing Machinery, **2012**: 1677–1681.

[126] J. Yang and J. Leskovec. "*Defining and Evaluating Network Communities Based on Ground-Truth*". In: **2015**: 181–213.

[127] A. Prat-Pérez, D. Dominguez-Sal and J.-L. Larriba-Pey. "*High quality, scalable and parallel community detection for large real graphs*". In: *Proceedings of the 23rd international conference on World wide web*. **2014**.

[128] J. Leskovec and A. Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection*. 2014-06.

[129] J. Leskovec and J. J. Mcauley. "*Learning to discover social circles in ego networks*". In: *Advances in neural information processing systems*. **2012**: 539–547.

[130] A. Lancichinetti, S. Fortunato and J. Kertész. "*Detecting the overlapping and hierarchical community structure in complex networks*". *New Journal of Physics*, **2009**, *11*(3): 033015.

[131] G. Palla, A.-L. Barabási and T. Vicsek. "*Quantifying social group evolution*". *Nature*, **2007**, *446*(7136): 664–667.

[132] M. E. Newman. "*Modularity and community structure in networks*". *Proceedings of the National Academy of Science*, **2006**, *103*(23): 8577–8582.

[133] D. Kempe, J. Kleinberg and É. Tardos. "*Maximizing the spread of influence through a social network*". In: *KDD'03*. **2003**: 137–146.

[134] Q. Mei, J. Guo and D. Radev. "*Divrank: the interplay of prestige and diversity in information networks*". In: *KDD'10*. **2010**: 1009–1018.

[135] M. S. Granovetter. "*The strength of weak ties*". In: *Social networks*. **1977**: 347–367.

[136] W. W. Powell, D. R. White, K. W. Koput *et al.* "*Network dynamics and field evolution: The growth of interorganizational collaboration in the life sciences*". *American journal of sociology*, **2005**, *110*(4): 1132–1205.

[137] H. R. Varian. *Intermediate Microeconomics: A Modern Approach*. WW Norton & Company, **2014**.

[138] H. Tong, J. He, Z. Wen *et al.* "*Diversified ranking on large graphs: an optimization viewpoint*". In: *KDD'11*. **2011**: 1028–1036.

[139] S. Gollapudi and A. Sharma. "*An axiomatic approach for result diversification*". In: *WWW'09*. **2009**: 381–390.

[140] A. Borodin, H. C. Lee and Y. Ye. "*Max-sum diversification, monotone submodular functions and dynamic updates*". In: *PODS'12*. **2012**: 155–166.

[141] N. Buchbinder, M. Feldman, J. S. Naor *et al.* "*Submodular maximization with cardinality constraints*". In: *SODA'14*. **2014**: 1433–1452.

[142] L. Page, S. Brin, R. Motwani *et al. The PageRank citation ranking: Bringing order to the web.* [techreport]. **1999**.

[143] J. He, H. Tong, Q. Mei *et al.* "*Gender: A generic diversified ranking algorithm*". In: *NIPS'12*. **2012**: 1142–1150.

[144] X. Zhu, A. Goldberg, J. Van Gael *et al.* "*Improving diversity in ranking using absorbing random walks*". In: *HLT-NAACL'07*. **2007**: 97–104.

[145] A. Dubey, S. Chakrabarti and C. Bhattacharyya. "*Diversity in ranking via resistive graph centers*". In: *KDD'11*. **2011**: 78–86.

[146] L. Du, Y. Wang, G. Song *et al.* "*Dynamic Network Embedding : An Extended Approach for Skip-gram based Network Embedding*". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, **2018**: 2086–2092.

[147] S. Chang, W. Han, J. Tang *et al.* "*Heterogeneous Network Embedding via Deep Architectures*". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, **2015**: 119–128.

[148] Y. Dong, N. V. Chawla and A. Swami. "*metapath2vec: Scalable representation learning for heterogeneous networks*". In: *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, **2017**: 135–144.

[149] M. Qu, J. Tang, J. Shang *et al.* "*An Attention-based Collaboration Framework for Multi-View Network Representation Learning*". In: *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*. New York, NY, USA: ACM, **2017**: 1767–1776.

[150] Anonymous. "*Data Poisoning Attack Against Node Embedding Methods*". In: *Submitted to International Conference on Learning Representations*. **2019**, under review.

[151] H. Dai, H. Li, T. Tian *et al.* "*Adversarial Attack on Graph Structured Data*". In: *Proceedings of the 35th International Conference on Machine Learning*. Stockholmsmässan, Stockholm Sweden: PMLR, 2018-07: 1115–1124.

[152] D. Zügner, A. Akbarnejad and S. Günnemann. "*Adversarial attacks on neural networks for graph data*". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. **2018**: 2847–2856.

[153] G. Frege. "*Über Sinn und Bedeutung*". *Zeitschrift für Philosophie und philosophische Kritik*, **1892**, *100*: 25–50.

[154] J. Chung, Ç. Gülçehre, K. Cho *et al.* "*Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*". In: *Proceedings of NIPS 2014 Deep Learning and Representation Learning Workshop*. **2014**.

[155] Y. LeCun, L. Bottou, Y. Bengio *et al.* "*Gradient-based learning applied to document recognition*". *Proceedings of the IEEE*, 1998-11, *86*(11): 2278–2324.

[156] R. Collobert, J. Weston, L. Bottou *et al.* "*Natural Language Processing (Almost) from Scratch*". *Journal of Machine Learning Research*, **2011**, *12*: 2493–2537.

[157] T. Mikolov, I. Sutskever, K. Chen *et al.* "*Distributed Representations of Words and Phrases and Their Compositionality*". In: *Proceedings of Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, **2013**: 3111–3119.

[158] C. Tu, H. Liu, Z. Liu *et al.* "*CANE: Context-Aware Network Embedding for Relation Modeling*". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*. Vancouver, Canada: Association for Computational Linguistics, **2017**: 1722–1731.

[159] S. Pan, J. Wu, X. Zhu *et al.* "*Tri-Party Deep Network Representation*". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence, **2016**: 1895–1901.

[160] T. Kenter, A. Borisov and M. de Rijke. "*Siamese CBOW: Optimizing Word Embeddings for Sentence Representations*". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, **2016**: 941–951.

[161] M. Kitsak, L. K. Gallos, S. Havlin *et al.* "*Identification of influential spreaders in complex networks*". *Nature physics*, **2010**, *6*(11): 888.

[162] G. F. De Arruda, A. L. Barbieri, P. M. Rodríguez *et al.* "*Role of centrality for the identification of influential spreaders in complex networks*". *Physical Review E*, **2014**, *90*(3): 032812.

[163] F. Chierichetti, S. Lattanzi and A. Panconesi. "*Rumour Spreading and Graph Conductance*". In: *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*. **2010**.

[164] O. Levy and Y. Goldberg. "*Neural word embedding as implicit matrix factorization*". In: *Advances in neural information processing systems*. **2014**: 2177–2185.

[165] Y. Li, L. Xu, F. Tian *et al.* "*Word Embedding Revisited: A New Representation Learning and Explicit Matrix Factorization Perspective*". In: *Proceedings of 24th International Joint Conference on Artificial Intelligence*. **2015**: 3650–3656.

[166] S. Wuchty and P. F. Stadler. "*Centers of complex networks*". *Journal of Theoretical Biology*, **2003**, *223*(1): 45–53.

[167] A. Mislove, H. S. Koppula, K. P. Gummadi *et al.* "*Growth of the flickr social network*". In: *Proceedings of the first workshop on Online social networks*. **2008**: 25–30.

[168] U. Brandes and C. Pich. "*Centrality estimation in large networks*". *International Journal of Bifurcation and Chaos*, **2007**, *17*(07): 2303–2318.

[169] A.-L. Barabási and R. Albert. "*Emergence of scaling in random networks*". *science*, **1999**, *286*(5439): 509–512.

[170] T. Mikolov, I. Sutskever, K. Chen *et al.* "*Distributed representations of words and phrases and their compositionality*". In: *Advances in neural information processing systems*. **2013**: 3111–3119.

[171] T. Hočevar and J. Demšar. "*A combinatorial approach to graphlet counting*". *Bioinformatics*, **2014**, *30*(4): 559–565.

[172] P. Yanardag and S. Vishwanathan. "*Deep Graph Kernels*". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Sydney, NSW, Australia, **2015**: 1365–1374.

[173] Ö. N. Yaveroğlu, N. Malod-Dognin, D. Davis *et al.* "*Revealing the hidden language of complex networks*". *Scientific reports*, **2014**, *4*: 4547.

[174] M. Jacomy, T. Venturini, S. Heymann *et al.* "*ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software*". *PloS one*, **2014**, *9*(6): e98679.

[175] R. Zafarani and H. Liu. *Social Computing Data Repository at ASU*. **2009**.

[176] C. Stark, B.-J. Breitkreutz, T. Reguly *et al.* "*BioGRID: a general repository for interaction datasets*". *Nucleic acids research*, **2006**, *34*(suppl 1): D535–D539.

[177] M. Mahoney. "*Large text compression benchmark*". **2009**.

[178] L. Tang and H. Liu. "*Leveraging social media networks for classification*". *Data Mining and Knowledge Discovery*, **2011**, *23*(3): 447–478.

[179] C. Donnat, M. Zitnik, D. Hallac *et al.* "*Learning Structural Node Embeddings via Diffusion Wavelets*". In: *International ACM Conference on Knowledge Discovery and Data Mining (KDD)*. **2018**.

[180] C. Gardiner. *Stochastic methods*. springer Berlin, **2009**.

[181] E. Lukacs. "*Characteristic functions*". **1970**.

[182] L. F. R. Ribeiro, P. H. P. Saverese and D. R. Figueiredo. "struc2vec*: Learning Node Representations from Structural Identity*". In: *The 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17*. **2017**: 385–394.

[183] F. Pedregosa, G. Varoquaux, A. Gramfort *et al.* "*Scikit-learn: Machine Learning in Python*". *Journal of Machine Learning Research*, **2011**, *12*: 2825–2830.

[184] L. He, C.-T. Lu, J. Ma *et al.* "*Joint Community and Structural Hole Spanner Detection via Harmonic Modularity*". In: *The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* **2016**: 875–884.

个人简历、在学期间的研究成果

.1 个人简历

1993 年出生于天津市；

2005 年至 2011 年就读于北京市十一学校；

2011 年至 2015 年就读于北京工业大学计算机学院，计算机科学与技术（实验班）专业，获理学学士学位；

2015 年保送进入北京大学信息科学技术学院，智能科学系，攻读博士学位至今。

.2 发表的学术论文

1. 社区发现 *Transactions on Social Computing* 已接收
   FOX: Fast Overlapping Community Detection Algorithm in Big Weighted Networks
   Tianshu Lyu, Lidong Bing, Zhao Zhang, Yan Zhang
2. 图表示学习 *PAKDD'20*
   Node Conductance: A Scalable Node Centrality Measure Based on Deepwalk
   Tianshu Lyu, Fei Sun, Yan Zhang
3. 图表示学习 *RecSys'19* EI: 20194207554865
   Compositional Network Embedding for Link Prediction
   Tianshu Lyu, Fei Sun, Peng Jiang, Wenwu Ou, Yan Zhang
4. 图表示学习 *CIKM'17* EI: 20175004524787
   Enhancing the Network Embedding Quality with Structural Similarity
   Tianshu Lyu, Yuan Zhang, Yan Zhang
5. 社区发现 *ICDM'16* EI: 20171003424001
   Efficient and Scalable Detection of Overlapping Communities in Big Networks
   Tianshu Lyu, Lidong Bing, Zhao Zhang, Yan Zhang
6. 推荐系统 *WISE'18* EI: 20184506032027
   PUB: Product Recommendation with Users' Buying Intents on Microblogs
   Xiaoxuan Ren, *Tianshu Lyu*, Zhao Zhang, Yan Zhang
7. 图表示学习 *AAAI'18* EI: 20190506436147
   COSINE: Community-Preserving Social Network Embedding from Information Diffusion Cascades

Yuan Zhang, *Tianshu Lyu*, Yan Zhang

8. **信息传播** *SIGIR'17* EI: 20173804170390
   Hierarchical Community-Level Information Diffusion Modeling in Social Networks
   Yuan Zhang, *Tianshu Lyu*, Yan Zhang

.3 参与编写的书籍

1. 《百面深度学习》，第三章图神经网络，ISBN: 9787115530974
   人民邮电出版社，诸葛越，江云胜主编

.4 参与的科研项目

1. 国家重点基础研究计划（973）项目“网络大数据计算的基础理论及其应用研究”之课题“网络大数据模式发现与效应分析方法研究”（课题号：2014CB340405）
2. 国家自然科学基金重点项目，面向课程的大规模在线教育资源组织与持续优化的理论与方法（课题号：61532001）
3. 教育部---中国移动科研基金项目，慕课教学效果与慕课的教育资源质量评价体系及应用研究（课题号：MCM20170503）

.5 所获奖励

1. 2016 — 2017 年度，北京大学优秀科研奖学金
2. 2017 — 2018 年度，北京大学专项奖学金

致谢

迄今为止，我想人生中最失落的时刻，就是 2011 年高考理综考试结束等待老师收卷的时候。当时心里清楚：无缘北大了。不过在这之前的 18 年和这之后的 9 年里，生活即使偶有波澜，总的来说却也算得上顺风顺水。以我对生活浅薄的见解，坎坷的出现大概率因个人的愚蠢产生，而平顺的维持却是时运与人和共同作用的结果。此篇致谢的意义即在于记录和感念我直博期间顺风顺水背后的人们。

2014 年准备保研的时候初识我的博士生导师，张岩教授。张老师慈眉善目，总是乐呵呵的，行事周全，思维又很跳跃，时常妙语连珠。在他的办公桌上，大概 80% 的面积都是一摞摞书堆起来的“屏风”，我恍惚记得有学科专业的大部头、计算机学会通讯，也有三联生活周刊、环球科学、人民文学，书铺得广，书的内容铺得更广。张老师的“有趣”可见一斑。在学术研究上，张老师耐心领我入门，之后既在点滴之中言传身教严谨治学的精神，又给我极大的自由探索不同的方向，研究的困难也总能在他的三言两语间疏解。老师的一言一行都体现着“心欲小而志欲大，智欲圆而行欲方，能欲多而事欲鲜”，这是我做张老师学生的 7 年间最深切的体会，也是我向往的处世哲学。

博士生期间，与李晓明教授的接触机会也颇多。他儒雅的风度，敏锐的思维，和积极投身教育事业的精神令我印象深刻并敬佩不已。李老师的《网络、人群与市场》课，课程内容和形式都有精巧设计，我受其影响也有意识地关注一些跨学科的研究。我想重要的不是知识本身，而是对世界保持赤子般的好奇与探索。在此也十分感谢李老师对我的博士论文给出中肯的意见。

在阿里巴巴淘宝信息流团队、hulu 推荐团队、Google GPay 团队，我有三段收获颇丰的实习经历。将在实验室的科研成果带到如此优秀的工业界团队是个令人兴奋的事情。除此以外，通过实习结识的几位 mentor 更令我常常感叹自己的幸运。阿里巴巴的孙飞，我戏称他为“远房师兄”，搜索推荐领域江湖百晓生，于我而言，是亦师亦友的角色。他帮助我的学术论文提供了思路和写作上的指导，在就业问题上亦给我很多真诚的建议。hulu 的祖松鹏、杨佳瑞手把手带我做算法工程师的工作，深度参与项目，是我进入工业界的引路人。他们二位都是清华非计算机专业的博士，却可以在算法工程师的岗位上游刃有余，想必是有过人的脑力和毅力。Google 的戴维与我的缘分更是深厚，他是毕业于北工大的“真师兄”。起初，我对 Google 偏向开发的实习工作并不熟悉，戴维颇具“大哥风度”，帮我扛了责任，推我在团队里展示自己，教我如何在顶尖互联网公司做个德智体美劳全面发展的程序员。倘若没有他的倾囊相授，我可能会在实习期间承受巨大的心理压力。临近毕业的这段时间里，几位 mentor 的无私帮助像明

灯照亮前路，让我在学术界和工业界之间的路上走得有底气。

几位师长与朋友为我的顺利保研和毕业也提供了很多帮助，一并向各位鞠躬致谢。准备保研时，我的哥哥周明昕搜集多方信息，让我能有机会提早了解智能系和 DAIR 组。本科的好友赵彤、徐舒怡、谢澜、张雨、张远行，我们有些在研究生期间继续做同学，实习的时候是同事，有些身在中美两地但依旧保持紧密的联系。秋招期间与李航博士、殷大伟博士、张富峥博士等有深入的交流，感谢他们对我的认可。

以我的观察，张老师 DAIR 实验室的风格甚为独特。在学术研究上，大家互相知晓研究方向和问题，但是具体的合作比较少，工作通常是独立完成。这客观上导致了大家之间的情谊十分纯粹，靠聚餐、出游、兴趣爱好、八卦轶事、插科打诨连结在一起。与我有较多接触的实验室成员们：张昭、年家震师兄、赵时师姐、高泽群师兄、林萍萍师姐、李鹏师兄、陈维政师兄、张元师弟、任笑萱师妹、刘志强师弟、张彧师弟、殷裔安师弟、李云涛师弟、付成真师妹、寇晓宇师妹、郭嘉炎师弟，大家性格迥异，但都是对生活有热情有态度的人，共同营造了理科楼 2216 轻松欢乐的氛围。特别感谢张昭在我初进组时带我吃遍北大周边所有餐馆，并将此项活动发展为每周一中午的全组聚餐。

感谢家人客观上和主观上带给我的一切。高中时期的某一天我偶然发现奶奶、爸爸和我都有相同的行为：睡觉的时候脚会伸到被子外面。这个发现带给我的震撼在于，除了样貌，我所以为的个人习惯和行为很多也是写在基因密码里的。民法规定：成年人为完全民事行为能力人，可以独立实施民事法律行为。开玩笑的讲，我的发现说明成年人的行为不是（统计意义上的）“独立”的。行为的不独立导致一个人的人生轨迹也受到来自家庭的难以磨灭的深远影响。言归正传，谢谢家人赋予我天生乐观又敏感的性格，谢谢家人给我尽可能最好的教育，全力的支持，与无限的爱。遗憾无法与一些家人分享此刻的喜悦，愿自己能在地球的土地上持续发光发热，与天上的星星相互照耀。

今年是我与沈学辉先生相识的第 12 年，期间经历了两人之间关系的种种变化，有细水长流的日常，也有戏剧化的桥段。感谢他的一路相伴。作为医药从业者，还要感谢他为人类生命健康作出的贡献！如今即将走向下一个里程碑，希望我们能活出一部接地气的喜剧。

正值新冠肺炎席卷全球，向医护工作者、基层服务人员、以及在艰难条件下保证社会生活基本运行的劳动者们表示衷心的感谢。