

图神经网络在复杂图挖掘上的研究进展*

刘杰, 尚学群, 宋凌云, 谭亚聪



(西北工业大学 计算机学院, 陕西 西安 710072)

通信作者: 尚学群, E-mail: shang@nwpu.edu.cn; 宋凌云, E-mail: lysong@nwpu.edu.cn

摘要: 图神经网络对非欧式空间数据建立了深度学习框架, 相比传统网络表示学习模型, 它对图结构能够实施更加深层的信息聚合操作. 近年来, 图神经网络完成了向复杂图结构的迁移, 诞生了一系列基于复杂图的图神经网络模型. 然而, 现有综述文章缺乏对复杂图神经网络全面、系统的归纳和总结工作. 将复杂图分为异质图、动态图和超图 3 种类型. 将异质图神经网络按照信息聚合方式划分为关系类型感知和元路径感知两大类, 在此基础上, 分别介绍普通异质图和知识图谱. 将动态图神经网络按照处理时序信息的方式划分成基于循环神经网络、基于自编码器以及时空图神经网络三大类. 将超图神经网络按照是否将超图展开成对图划分为展开型和非展开型两大类, 进一步按照展开方式将展开型划分成星形展开、团式展开和线形展开 3 种类型. 详细阐述了每种算法的核心思想, 比较了不同算法间的优缺点, 系统列举了各类复杂图神经网络的关键算法、(交叉)应用领域和常用数据集, 并对未来可能的研究方向进行了展望.

关键词: 图神经网络; 复杂图; 异质图; 动态图; 超图

中图法分类号: TP18

中文引用格式: 刘杰, 尚学群, 宋凌云, 谭亚聪. 图神经网络在复杂图挖掘上的研究进展. 软件学报, 2022, 33(10): 3582–3618. <http://www.jos.org.cn/1000-9825/6626.htm>

英文引用格式: Liu J, Shang XQ, Song LY, Tan YC. Progress of Graph Neural Networks on Complex Graph Mining. Ruan Jian Xue Bao/Journal of Software, 2022, 33(10): 3582–3618 (in Chinese). <http://www.jos.org.cn/1000-9825/6626.htm>

Progress of Graph Neural Networks on Complex Graph Mining

LIU Jie, SHANG Xue-Qun, SONG Ling-Yun, TAN Ya-Cong

(School of Computer Science, Northwestern Polytechnic University, Xi'an 710072, China)

Abstract: Graph neural networks (GNNs) establish a deep learning framework for non-Euclidean spatial data. Compared with traditional network embedding methods, they perform deeper aggregating operations on graph structures. In recent years, GNNs have been extended to complex graphs. Nevertheless, there lacks qualified surveys which give comprehensive and systematic classification and summary on GNNs based on complex graphs. This study divides the complex graphs into 3 categories, i.e., heterogeneous graphs, dynamic graphs, and hypergraphs. GNNs based on heterogeneous graphs are divided into 2 types, i.e., edge-type aware and meta-path aware, according to the procedure that the information is aggregated. Dynamic GNNs graphs are divided into three categories: RNN-based methods, autoencoder-based methods, and spatio-temporal graph neural networks. Hypergraph GNNs are divided into expansion methods and non-expansion methods, and the expansion methods are further divided into star-expansion, clique-expansion, and line-expansion according to the expansion mode they use. The core idea of every method is illustrated in detail, the advantages and disadvantages of different algorithms are compared, the key procedures, (cross) application fields, and commonly used data sets of different complex graph GNNs are systematically listed, and some possible research directions are proposed.

Key words: graph neural network (GNN); complex graph; heterogeneous graph; dynamic graph; hypergraph

* 基金项目: 国家重点研发计划(2020AAA0108504); 国家自然科学基金(62102321, 61772426, U1811262); 中央高校基本科研专项基金(D5000200146); 中国博士后科学基金(2020M673487)

本文由“智慧信息系统新技术”专题特约编辑邢春晓研究员、王鑫教授、张勇副研究员、于戈教授推荐.

收稿时间: 2021-07-19; 修改时间: 2021-08-30; 采用时间: 2021-12-24; jos 在线出版时间: 2022-02-22

1 背景与分类

图(graph)作为一种数据结构,能够精确描述事物间复杂的相互作用关系,因而被广泛应用于诸多科学和工程领域.近年来,由于图结构丰富的信息表达能力,利用机器学习对图进行分析的研究受到越来越多的关注.图表示学习,便是将图中丰富的结构和语义信息转化成低维稠密的节点表示向量,以便于后续利用机器学习方法进行诸如节点分类、链接预测和知识推断等图相关应用.其中,图神经网络(graph neural networks, GNNs)由于对图结构建立了深度学习框架,相比于 DeepWalk^[1]、Node2vec^[2]等传统网络表示学习方法,可以同时利用图结构信息和节点特征信息,并构造更加复杂深层的神经网络进行表示学习,因而逐渐成为了近年来的研究热点.

为了简化问题和便于建模,早期的图神经网络大多基于简单图结构,即静态、同质的成对图结构. Sperduti 等人^[3]最早尝试将神经网络应用到有向无环图上.他们提出了一个将输入的图结构转化成固定维度的节点向量的编码器,并将向量输入到一个前馈神经网络中进行分类.在此基础上, Gori 等人^[4]首次提出了图神经网络的概念,他们将递归神经网络(recursive neural network, RNN)扩展到了图结构,进而提出了递归图神经网络.由此衍生出一系列早期图神经网络研究^[5,6].然而,这些早期方法有着很高的计算复杂度,难以应用到大型图结构上,因此并没有得到广泛应用.

近年来,得益于卷积神经网络(convolutional neural network, CNN)在计算机视觉领域的成功,许多方法开始尝试对图结构定义卷积操作,并将卷积神经网络迁移到图结构上.这些方法统称为图卷积神经网络(graph convolutional networks, GCNs).图卷积神经网络可以按照实施卷积的方式分为谱方法(spectral-based GCNs)和空间方法(spatial-based GCNs)两种类型.其中,谱方法利用卷积定理从谱域定义图卷积.例如, Xu 等人提出的 GWNN^[7]引入小波变换替换傅里叶变换作为基底,使得模型拥有更好的局部性; Defferrard 等人^[8]和 Kipf 等人^[9]通过对卷积核参数化,实现了局部性并降低了复杂度.空间方法则致力于从节点所在的空间域出发,通过定义聚合操作和连接操作来聚合邻居信息并与中心节点信息合并从而形成新的中心节点表示.例如, MPNN^[10]将图卷积转化成节点间信息的传递,并提出了空间方法的一个通用框架.后续的 GraphSage^[11]使用采样选取固定数量的邻居节点,并给出更加丰富的聚合函数类型.

随着 GNNs 在简单图上的逐渐完善,人们开始考虑更加复杂多样的图结构.通过赋予图异质性、时序性或高阶关联等特征, GNNs 衍生出了一大批基于复杂图结构的变种,使其拥有更加灵活广泛的应用场景.本文按照复杂图的结构类型将这些方法分为异质图、动态图和超图三大类.图 1 表示了简单图和 3 种复杂图结构.

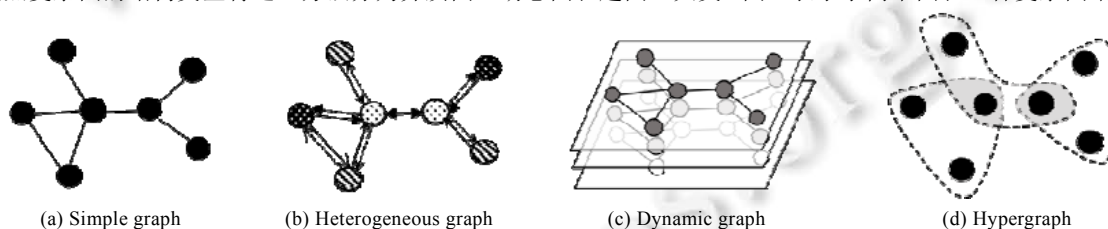


图 1 简单图和 3 种复杂图结构

(1) 异质图神经网络

在真实世界中,事物之间的相互作用关系构成的图结构往往是异质的,即节点和边具有多种类型.例如,在社交网络中同时存在用户、贴文和评论等类型的节点和用户-贴文、用户-评论和贴文-评论等多种关系.边和节点的异质造成了节点间语义关系的多样性,同时,边和节点所具有的属性特征也位于不同的特征空间中,这些原因导致传统的图神经网络方法无法高效、准确地处理异质图.为了解决这些问题, Schlichtkrull 等人^[12]最早通过对不同类型的关系定义不同的系数矩阵,将 GCN^[9]成功迁移到了异质图上; Hu 等人^[13]在此基础上,进一步使用注意力机制计算不同类型关系的重要程度;后续的王ang 等人^[14]则引入了元路径概念将异质图转化为同质图,再进行表示学习.这些方法被统一归纳为异质图神经网络(heterogeneous graph neural networks, HetGNNs).

(2) 动态图神经网络

在真实世界中, 图的结构信息和节点特征会随时间发生变化, 从而导致对图神经网络的输入发生变化. 这类复杂图出现在交通网络、生物网络及知识图谱等各种应用中. 例如, 在时序链接预测问题上, 模型需要根据网络在 0 到 t 时刻的变化情况预测 $t+1$ 时刻的网络状态; 在交通流预测问题中, 某一位置的交通状态既受到邻近位置交通流的影响, 也受到这些位置历史交通状态的影响. 因此, 为了在实施图卷积的过程中充分考虑图结构和节点属性的动态特性, 一些研究者^[15,16]开始将循环神经网络与 GNNs 相结合, 或引入自编码器来处理动态图^[17]. 此外, Yan 等人^[18]将时序关系转化成时间连接, 提出了时空图神经网络. 这些方法都被归纳为动态图神经网络(dynamic graph neural networks, DGNNs).

(3) 超图神经网络

在现实世界中, 节点间的关系常常不是成对出现的, 而是两个或两个以上节点间共同构成相互作用关系. 例如, 在生物蛋白质交互网络中, 往往是多个蛋白质共同作用行使某项生物功能; 在引文网络中, 往往是多位作者共同写作了某篇文章. 这些高阶关联形成了一个完整的关系整体, 因而将这种高阶关联使用成对图结构分别表示将会带来信息损失^[19]. 超图(hypergraph)扩展了图的定义, 其中的一条超边(hyperedge)可以包含任意数量的节点, 因此可以直接储存高阶关联. 近年来, 超图的优势引起了学术界和工业界的广泛关注, 一系列超图神经网络方法(hypergraph neural network, HyperGNN)被相继提出. 按照是否将超图展开为成对图, 这些方法可分为展开型^[20,21]和非展开型^[22,23]超图神经网络. 图 2 展示了本综述对于图神经网络方法的分类框架.

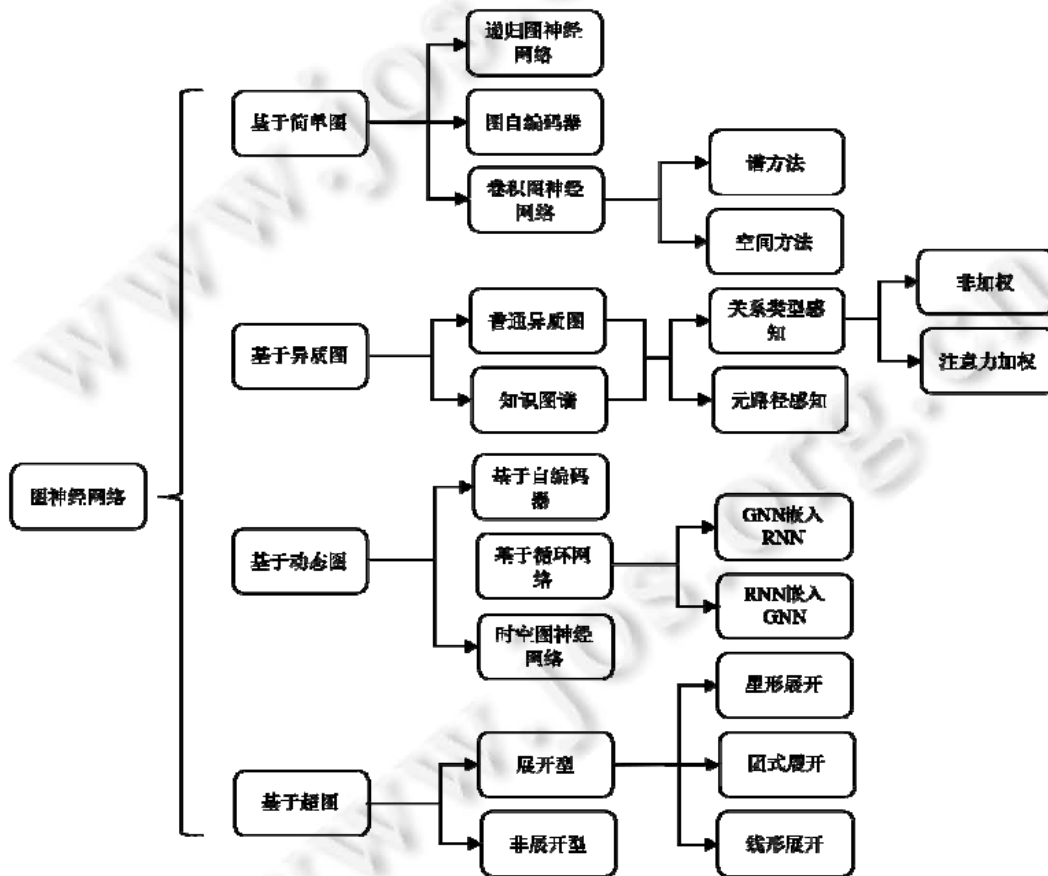


图 2 基于图类型的 GNN 分类

2 相关综述工作

近年来, 由于强大的性能和广泛的应用场景, 神经网络逐渐成为一项研究热点, 相关研究层出不穷. 为了有效整理研究现状, 为后续研究者提供参考和检索, 不少神经网络的综述性文章陆续发表出来. 其中, Wu 等人^[24]针对现有的神经网络方法提出了一个基础分类框架, 将现有算法分成了 4 种类型: 递归神经网络、卷积神经网络、图自编码器和时空神经网络, 并讨论了常用的基准数据集以及神经网络在各个领域中的应用. Zhang 等人^[25]则按照递归神经网络、卷积神经网络、图自编码器、图强化学习和图生成对抗学习这 5 个方面进行综述. Zhou 等人^[26]按照前向传播模型、采样模型和池化模型对神经网络进行分类, 并简要介绍了基于复杂图结构的神经网络方法. Jiang 等人^[27]针对神经网络在交通流预测方面的应用进行了总结和归纳. Wu 等人^[28]则关注了神经网络在推荐系统上的应用.

尽管现在已经存在许多关于神经网络方法或应用的综述, 但它们均存在以下缺陷: (1) 当前综述包含的神经网络主要基于简单图结构(同质、静态的成对图), 然而近年来, 神经网络已经完成了向异质图、动态图和超图等复杂图结构的迁移, 并取得了优秀的成果, 这些方法并未得到很好的归纳总结. 例如, Zhou 等人^[26]的综述虽然提到了一些基于时序图或超图的 GNN 方法, 但不够系统和全面, 很多最新研究没有被涉及. (2) 现有综述对复杂图 GNN 缺乏全面清晰的分类框架, 例如, Wu 等人^[24]的综述在归纳动态图 GNN 时只涉及了时空神经网络, 而时空图只是动态图的一种. (3) 上述 3 种复杂图可以进一步结合形成诸如动态知识图谱或异质超图等更加复杂的结构, 现有综述对基于这些图结构的 GNN 以及相关的应用并没有很好的归纳总结.

综上所述, 目前仍缺乏针对神经网络在多种复杂图结构上全面、成体系的分类和总结工作. 为此, 本文针对异质图、动态图和超图等复杂图结构的神经网络方法提出了全面的分类框架并进行了系统的总结, 包含了大量经典方法和最新的研究进展.

本文第 3 节给出几种复杂图的形式化定义并列举涉及到的变量符号. 第 4 节简要介绍简单图 GNN, 并进行分类和比较. 第 5-7 节分别对异质图 GNN、动态图 GNN 以及超图 GNN 进行详细阐述以及分类比较. 第 8 节介绍 3 种复杂图各自的应用领域以及组合后的应用. 第 9 节给出几种复杂图的常用数据集. 第 10 节对潜在的研究方向进行展望和总结.

3 概念与定义

本节我们将给定几种复杂图结构的符号化定义. 其中, 大写加粗字母表示矩阵, 小写加粗字母表示向量.

定义 1(简单图). 在本文中我们定义同质、静态的成对关系图为简单图结构, 可表示为 $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. 其中, $\mathcal{V} = \{v_1, \dots, v_n\}$ 表示图中的节点集合, $\mathcal{E} = \{e_1, \dots, e_m\}$ 表示边的集合. 对任意边, $e_{ij} \in \mathcal{E}$, $e_{ij} = (v_i, v_j) \in \mathcal{V}$. 图结构信息可用邻接矩阵 $\mathbf{A} \in \mathbb{R}^{n \times n}$ 表示 $\mathbf{X}^e \in \mathbb{R}^{|\mathcal{E}| \times d}$:

$$\mathbf{A}(i, j) = \begin{cases} 1, & \text{if } v_i \in \mathcal{E} \\ 0, & \text{if } v_i \notin \mathcal{E} \end{cases}$$

图中节点可能具有属性特征 \mathbf{X} , 其中, $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ 表示节点特征矩阵, $\mathbf{x}_i = \mathbf{X}[i, :] \in \mathbb{R}^d$ 表示节点 v_i 的特征向量. 图中的边也可能具有属性特征 \mathbf{X}^e , 其中, $\mathbf{X} \in \mathbb{R}^{|\mathcal{E}| \times d}$ 表示超边特征矩阵, $\mathbf{x}_j^e = \mathbf{X}[j, :] \in \mathbb{R}^d$ 表示超边 e_j 的特征向量.

定义 2(异质图). 异质图中节点和边可以具有不同类型, 因此, 异质图可定义为 $\mathcal{G}^h = \{\mathcal{V}^h, \mathcal{E}^h, \mathcal{B}_v, \mathcal{B}_e\}$. 其中, \mathcal{V}^h 代表节点集合, \mathcal{E}^h 代表边的集合, \mathcal{B}_v 表示节点种类集合, \mathcal{B}_e 表示边的种类集合. 当 $|\mathcal{B}_v| + |\mathcal{B}_e| = 2$ 时, \mathcal{G}^h 退化成同质图 \mathcal{G} ; 当 $|\mathcal{B}_v| + |\mathcal{B}_e| > 2$ 时, \mathcal{G}^h 表示异质图. 由于节点和边的异质性, 不同类型节点或边的属性特征可能落在不同特征空间, 节点特征矩阵表示为 $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{|\mathcal{B}_v|}\}$, 边的特征矩阵表示为 $\{\mathbf{X}_{e_1}^e, \mathbf{X}_{e_2}^e, \dots, \mathbf{X}_{e_{|\mathcal{B}_e|}}^e\}$. 异质图中经常出现元路径(meta-path)这一概念, 元路径 ξ 是异质图中任意一对节点间的连接路径, 描述了两个节点间的复合关系: $\xi = v_1 \xrightarrow{r_1} v_2 \xrightarrow{r_2} \dots \xrightarrow{r_k} v_{k+1}$, 其中, r_i 表示节点与 v_i 之间的关系.

定义 3(动态图). 动态图是一种结构信息和节点属性会随时间发生变化的图结构, 可定义为 $\mathcal{G}^{(t)} = \{\mathcal{V}^{(t)}, \mathcal{E}^{(t)}, \mathbf{X}^{(t)}\}$, 其中, $\mathbf{X}^{(t)} \in \mathbb{R}^{|\mathcal{V}^{(t)}| \times d}$ 表示节点属性信息. 时空图是一种特殊的动态图, 它连接了不同快照中的相同节点, 因此既拥有快照内节点的空间连接, 又具有快照间节点的时间连接.

定义 4(超图). 超图结构可定义为 $\mathcal{G}^p = \{\mathcal{V}^p, \mathcal{E}^p\}$, 其中, \mathcal{V}^p 代表节点集合, \mathcal{E}^p 代表超边集合. 在超图中, 允许一条超边同时连接两个以上的节点, 即对任意超边 $e \in \mathcal{E}^p$, $e = \{v_i, v_j, \dots, v_k\} \subseteq \mathcal{V}^p$. 每条超边可以定义权重 w_i , 我们使用正值对角矩阵 $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}^p| \times |\mathcal{V}^p|}$ 表示超图内所有超边的权重. 超图结构信息可以用关联矩阵 $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}^p| \times |\mathcal{E}^p|}$ 表示, 具体地:

$$\mathbf{H}(i, j) = \begin{cases} 1, & \text{if } v_i \in e_j \\ 0, & \text{if } v_i \notin e_j \end{cases}$$

超图中某条超边 e_j 可能同时包含多个节点 $\{v_i, v_j, \dots, v_k\}$. 同时, 某个节点 v_i 可能同时落在多条超边 $\{e_1, e_2, \dots, e_j\}$ 当中, 因此我们定义节点的度矩阵 $\mathbf{D}_v \in \mathbb{R}^{|\mathcal{V}^p| \times |\mathcal{V}^p|}$ 和超边度矩阵 $\mathbf{D}_e \in \mathbb{R}^{|\mathcal{E}^p| \times |\mathcal{E}^p|}$. \mathbf{D}_v 和 \mathbf{D}_e 都是对角矩阵. 其中, $\mathbf{D}_v(i, i) = \sum_{e \in \mathcal{E}^p} \mathbf{W}(e) \mathbf{H}(i, e)$, $\mathbf{D}_e(i, i) = \sum_{v \in \mathcal{V}^p} \mathbf{H}(v, i)$.

图神经网络常用的符号和相应描述见表 1.

表 1 GNN 常用符号与描述

符号	描述	符号	描述
$\mathcal{G} / \mathcal{G}^h / \mathcal{G}^{(t)} / \mathcal{G}^p$	简单图/异质图/时空图/超图	$\mathcal{V} / \mathcal{V}^p / \mathcal{V}^{(t)} / \mathcal{V}^p$	节点集合
$\mathcal{E} / \mathcal{E}^h / \mathcal{E}^{(t)} / \mathcal{E}^p$	边/超边集合	e	边/超边, $e \in \mathcal{E}$
v	节点, $v \in \mathcal{V}$	$ \cdot $	集合的模
\odot	逐元素求积	$\mathbf{A} \in \mathbb{R}^{n \times n}$	邻接矩阵
$\mathbf{H} \in \mathbb{R}^{ \mathcal{V}^p \times \mathcal{E}^p }$	超图关联矩阵	$\mathbf{W} \in \mathbb{R}^{ \mathcal{V}^p \times \mathcal{V}^p }$	超边权重矩阵
$\mathbf{D}_v \in \mathbb{R}^{ \mathcal{V}^p \times \mathcal{V}^p }$	超图节点度矩阵	$\mathbf{D}_e \in \mathbb{R}^{ \mathcal{E}^p \times \mathcal{E}^p }$	超边度矩阵
$\mathbf{X} \in \mathbb{R}^{ \mathcal{V} \times d}$	节点属性特征矩阵	$\mathbf{x}_i \in \mathbb{R}^d$	节点特征向量, $\mathbf{x}_i = \mathbf{X}[i, :]$
$\mathbf{X}^e \in \mathbb{R}^{ \mathcal{E} \times d}$	边属性特征矩阵	$\mathbf{X}_j^e \in \mathbb{R}^l$	边特征向量, $\mathbf{X}_j^e \in \mathbf{X}[j, :]$
\mathcal{B}_v	节点类型集合	\mathcal{B}_e	边类型集合
ξ	元路径	\parallel	连接操作

4 简单图神经网络

为了简化问题, 便于建模, 早期图神经网络的研究多基于简单图结构. 这一时期的研究重点集中在如何将深度学习方法(RNN、CNN、Autoencoder 等)应用到图结构学习当中, 并衍生出不同的发展路线. 基于简单图的图神经网络是基于复杂图的图神经网络的基础, 为了方便读者理解, 本节我们将简要介绍简单图神经网络. 我们将其大致分成 3 种类型: 递归图神经网络、卷积图神经网络和图自编码器, 并分别进行阐述.

4.1 递归图神经网络

递归图神经网络是最早尝试将深度学习迁移到图结构的方法. 它们利用图中的节点和边的信息来训练递归神经网络中一组固定的参数, 从而将节点周围的邻居信息融入到节点表示当中. Scarselli 等人^[6]最早将递归神经网络(recurrent neural network, RNN^[29])扩展到简单图结构并提出了 GNN^[6]. GNN 循环地将邻居节点和链接信息融合到中心节点上, 直到图中所有节点达到平衡状态. 具体地, 某个节点 i 的状态更新过程如下:

$$\mathbf{h}_i^{(l)} = \sum_{j \in \mathcal{N}(i)} \mathcal{F}(\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}, \mathbf{x}_i, \mathbf{x}_{(i,j)}^e, \mathbf{x}_j) \quad (1)$$

其中, $\mathcal{F}(\cdot)$ 表示需要学习到参数函数, $\mathbf{h}_i^{(l)}$ 表示第 l 层隐藏层状态, $\mathbf{h}_i^{(0)}$ 通过随机初始化得到.

Scarselli 等人^[6]提出了一个半监督方法来学习上述模型的参数. 首先, 使用雅各比方法^[30]不断迭代公式(1)直到节点隐藏向量达到稳定状态. 在得到稳定的 $\mathbf{h}_i^{(l)}$ 之后, GNN 通过参数函数 $\mathcal{O}(\cdot)$ 得到输出层:

$$\hat{y}_i = \mathcal{O}(\mathbf{h}_i^{(l)}, \mathbf{x}_i) \quad (2)$$

GNN 接下来通过 Almedia-Pineda 算法^[31]进行梯度下降, 从而最小化目标函数. 目标函数由具体任务决定, 例如在回归任务中, 目标函数可以是预测值与真实值之间的均方损失. 通过上述过程, GNN 可以有效学习有向无环图的结构信息. 尽管 GNN 具有重要意义, 但由于需要不断迭代模型直至达到稳定的节点状态, 它具有很高的时间复杂度. 为了解决这个问题, Gallicchio 等人提出了 GraphESN^[5]以提高 GNN^[6]的训练效率. GraphESN^[5]由一个编码器和一个输出层组成. 编码器被随机初始化并且不需要训练. 它实现了一个收缩状态转换函数来循环更新节点的状态, 直到全局图状态达到收敛. 之后, 输出层通过将固定节点的状态作为输入进行训练.

另一项针对 GNN^[6]的改进是 Li 等人提出的门控图序列神经网络 GGNNs^[32]. GGNNs 通过引入门控递归单元(gated recursive units, GRU^[33])来替换公式(1)中的循环迭代, 使得模型可以通过更加快速的优化方式进行训练. 具体地, 公式(1)被改进成以下形式:

$$\mathbf{h}_i^{(l)} = (1 - \mathbf{z}_i^{(l)}) \odot \mathbf{h}_i^{(l-1)} + \mathbf{z}_i^{(l)} \odot \hat{\mathbf{h}}_i^{(l)} \quad (3)$$

其中, $\mathbf{z}_i^{(l)}$ 表示门控向量, $\hat{\mathbf{h}}_i^{(l)}$ 表示待更新的隐藏状态. 与 GNN^[6]和 GraphESN^[5]不同, GGNNs 使用了反向传播算法来学习模型参数, 但对于较大的图结构, 模型复杂度仍会很高.

随机稳态嵌入算法 SSE^[34]采用了与 GGNNs 相似的框架, 但不同的是, SSE 采用了基于固定点的随机梯度下降方法来加速训练过程. 这个加速方法的基本思路是交替进行稳定节点状态计算与模型参数优化两个步骤, 并同时在二者上使用随机小批量训练.

递归神经网络作为早期的图神经网络, 尽管由于较高的复杂度以及囿于当时的计算能力并未引起太大关注, 但对后续研究产生了深远影响. GNN^[6]等递归神经网络方法循环地将邻居节点信息融合到中心节点, 这与后续提出的图卷积算子^[9]和聚合函数^[11]的思路非常相似. 值得注意的是, 近年来, 一些研究将目光对准了使用 RNN 训练动态图^[16,35]. 这些方法不再使用 RNN 来学习单个节点的隐藏状态, 而是使用 RNN 对不同时间点下的整个图结构进行编码. 我们将这类方法归类于基于动态图的图神经网络, 并在第 6 节给出详细叙述.

4.2 图自编码器

不同于第 4.3 节介绍的卷积图神经网络, 图自编码器并未延续早期循环图神经网络的研究思路, 而是源于与图神经网络有着紧密关联的另一类研究: 网络表示学习(network embedding). 网络表示学习是指学习网络中节点的低维表示向量的方法的总称. 尽管图神经网络和网络表示学习对图结构有着相似的学习目标(学习节点低维向量)和高度重合的下游工作(节点分类, 链接预测等), 二者在学习表示向量的过程中主要有两点区别: (1) 图神经网络主要基于深度学习模型, 而网络表示学习方法则涉及更加广泛的模型类型, 例如, 矩阵分解^[36]和随机游走^[1]在网络表示学习中也被广泛使用; (2) 许多网络表示学习方法具有邻居阶数的限制, 而图神经网络由于显式定义了图结构的深度学习框架, 因而被认为可以学习到更加高阶的节点表示. 图自编码器的提出源自于网络表示学习的研究, 但因为使用了深度学习框架, 我们将它也归结为一种图神经网络.

图自编码器一般由编码器和解码器两部分组成. 编码器是将图中节点信息映射到潜在的特征空间, 从而得到节点的隐藏表示; 而解码器则将节点特征重新编码成邻接矩阵或其他信息矩阵, 使得节点特征包含了图的结构信息.

Wang 等人提出的 SDNE^[37]率先引入了深度自编码器来学习网络的结构信息. 具体来说, SDNE 基于相似性假设学习节点的表示向量, 即认为在网络中拥有相似结构的节点相应的表示向量也应相近. 为了充分衡量节点间的相似性, SDNE 同时利用了一阶相似性(节点之间的相似性)和二阶相似性(节点邻域间的相似性)共同作为损失函数. 具体地, 给定节点 i 在邻接矩阵 \mathbf{A} 中的行向量 \mathbf{a}_i , SDNE 通过一个深度自编码器得到节点的隐藏层表示 $\mathbf{z}_i^{(k)}$:

$$\mathbf{z}_i^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{z}_i^{(l-1)} + \mathbf{b}^{(l)}), l=1,2,\dots,L \quad (4)$$

其中, $\mathbf{z}_i^{(0)} = \mathbf{a}_i$, 为了使节点隐藏表示满足一阶相似性, 定义一阶损失函数如下:

$$\mathcal{L}_{1st} = \sum_{i,j=1}^n a_{i,j} \|\mathbf{z}_i^{(L)} - \mathbf{z}_j^{(L)}\|_2^2 \quad (5)$$

其中, $a_{i,j}$ 表示节点 i, j 在邻接矩阵中的值. 一阶相似度使得在网络中直接相连的节点具有相近的表示向量. 然而, 图结构具有稀疏性, 许多节点间并没有直接相连, 但仍可能具有相似性, 因此, \mathcal{L}_{1st} 的约束性有限. 作为一阶相似性的补充, SDNE^[37] 引入了二阶损失函数. 具体地, 它通过一个深度解码器将节点隐藏向量恢复成邻接矩阵 $\hat{\mathbf{A}}$. 通过定义重构的邻接矩阵 $\hat{\mathbf{A}}$ 和原始邻接矩阵 \mathbf{A} 间的误差作为损失函数, 引入了节点邻域的相似性作为约束. 二阶损失函数形式如下:

$$\mathcal{L}_{2nd} = \sum_{i=1}^n \|(\hat{\mathbf{a}}_i - \mathbf{a}_i) \odot \mathbf{b}_i\|_2^2 \quad (6)$$

其中, $\mathbf{b}_i = \{b_{i,j}\}_{j=1}^n$, 如果 $a_{i,j} = 0$, $b_{i,j} = 1$; 如果 $a_{i,j} = 1$, $b_{i,j} = \beta$. 引入 \mathbf{b}_i 可以对非零项分配更高的权重. Cao 等人提出的 DNGR^[38] 也采用了自编码器学习节点表示, 但将邻接矩阵替换成了正点互信息(positive pointwise mutual information, PPMI)矩阵. 注意到 SDNE^[37] 和 DNGR^[38] 只考虑了网络结构信息, 而未将节点本身的特征融合到表示向量中, GC-MC^[39] 使用 Kipf 等人提出的 GCN^[9] 代替了多个全连层作为编码器. 这样一来, 模型可以同时考虑图结构信息和节点特征信息:

$$\mathbf{Z} = f(\mathbf{X}, \mathbf{A}) = \sigma_1(\mathbf{A}\sigma_2(\mathbf{A}\mathbf{X}\mathbf{W}^{(0)})\mathbf{W}^{(1)}) \quad (7)$$

其中, \mathbf{Z} 表示节点表示向量矩阵, σ_i 表示激活函数. 同时, GC-MC 使用一个双线性模型作为解码器:

$$\hat{\mathbf{A}}(i, j) = \text{softmax}(\mathbf{z}_i \mathbf{Q}_r \mathbf{z}_j^T) \quad (8)$$

其中, \mathbf{Q}_r 表示可学习的参数矩阵, \mathbf{z}_i 和 \mathbf{z}_j 表示节点 i 和 j 的表示向量. GC-MC 使用负交叉熵作为损失函数.

变分自编码器(variational autoencoder, VAE)是另一种自编码器框架, 不同于 AE, VAE 将隐藏层向量视为服从正态分布的随机变量而非固定值. 这样可以使得编码器具有更强的鲁棒性, 隐藏层表示也更加平滑^[40]. Kipf 等人首次将变分自编码器扩展到图结构并提出了变分图自编码器 VGAE^[41]. VGAE 假设每个节点的隐藏表示向量 \mathbf{z}_i 服从正态分布, 即 $q(\mathbf{z}_i | \mathbf{X}, \mathbf{A}) = \mathcal{N}(\mathbf{z}_i | \boldsymbol{\mu}, \text{diag}(\boldsymbol{\sigma}_i^2))$. 因此学习节点隐藏向量 \mathbf{z}_i 转换成了学习它的均值 $\boldsymbol{\mu}_i$ 和方差 $\boldsymbol{\sigma}_i^2$. 与 GC-MC^[39] 相似, VGAE 使用了一个两层 GCN 作为编码器:

$$\boldsymbol{\mu} = \text{GCN}_{\boldsymbol{\mu}}(\mathbf{X}, \mathbf{A}), \log \boldsymbol{\sigma} = \text{GCN}_{\boldsymbol{\sigma}}(\mathbf{X}, \mathbf{A}) \quad (9)$$

对于解码器, VGAE 使用了节点隐藏向量间的内积作为重构邻接矩阵的预测值:

$$p(\mathbf{A} | \mathbf{Z}) = \prod_{i=1}^N \prod_{j=1}^N p(A_{ij} | \mathbf{z}_i, \mathbf{z}_j), \quad p(A_{ij} = 1 | \mathbf{z}_i, \mathbf{z}_j) = \sigma(\mathbf{z}_i^T \mathbf{z}_j) \quad (10)$$

其中, \mathbf{A}_{ij} 是邻接矩阵 \mathbf{A} 的一项, $\sigma(\cdot)$ 表示 sigmoid 激活函数. 通过优化变分下界, 得到 VGAE 的损失函数:

$$\mathbf{L} = \mathbb{E}_{q(\mathbf{Z} | \mathbf{X}, \mathbf{A})} [\log_p(A | \mathbf{Z})] - \text{KL}[q(\mathbf{Z} | \mathbf{X}, \mathbf{A}) \| p(\mathbf{Z})] \quad (11)$$

其中, $\text{KL}[\cdot]$ 是 KL 散度, 用于衡量经验分布 $q(\cdot)$ 和先验分布 $p(\cdot)$ 间的距离, 而先验分布 $p(\mathbf{Z})$ 被定义为 $p(\mathbf{Z}) = \prod_i p(\mathbf{z}_i) = \prod_i \mathcal{N}(\mathbf{z}_i | \mathbf{0}, \mathbf{I})$. VGAE 引入 KL 散度的目的是要使得通过训练得到的经验分布 $q(\mathbf{Z} | \mathbf{X}, \mathbf{A})$ 与先验分布尽可能地相近. Pan 等人提出的 ARVGA^[42] 延续了这个思路, 并使用了生成对抗网络(generative adversarial networks, GAN^[43])作为编码器, 从而使得经验分布和先验分布尽可能地无法辨识. 除了上述介绍的 GAE 方法, DRNE^[44] 引入 LSTM^[45] 处理邻居节点序列并以此作为该节点的隐藏表示向量; NetRA^[46] 同样引入了生成对抗机制, 并利用 LSTM 对通过随机游走获得的邻居节点序列进行编码.

图自编码器利用深度自编码器对图结构数据进行表示学习, 通过定义不同的编码器与解码器形式, 它可以灵活适用多种图学习任务. 引入 GCN 作为编码器也解决了全连层编码无法融入节点特征的缺陷. 图自编码器是当前图神经网络研究的热点之一.

4.3 卷积神经网络

卷积神经网络可以分为谱方法和空间方法两类. 谱方法通过傅里叶变换或其他方式将节点变换到谱域

来执行卷积操作, 而空间方法通过考虑节点邻域信息进行信息聚合.

4.3.1 谱方法

谱方法是经过严密数学推导的一类方法, 它基于无向图进行研究. 无向图的结构信息可以用正则化拉普拉斯矩阵 $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ 表示, 其中, \mathbf{D} 是表示节点度的对角矩阵. \mathbf{L} 是实对称半正定矩阵, 因而可以分解为 $\mathbf{L} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T$, 其中, \mathbf{U} 是 \mathbf{L} 的特征向量构成的矩阵, $\mathbf{\Lambda}$ 是特征值构成的对角矩阵. 基于图的傅里叶变换便是将节点特征通过特征矩阵 \mathbf{U} 投影到谱域. 具体地, 对于节点特征向量 $\mathbf{x} \in \mathbb{R}^d$, 傅里叶变换定义为 $\mathcal{F}(\mathbf{x}) = \mathbf{U}^T \mathbf{x}$, 相应可定义傅里叶逆变换为 $\mathcal{F}^{-1}(\mathbf{x}') = \mathbf{U} \mathbf{x}'$, 其中, \mathbf{x}' 是变换后的节点特征. 最后, 为了进行卷积操作, 过滤器被定义为 $g_\theta \in \mathbb{R}^d$. 图卷积过程定义如下:

$$g_\theta * \mathbf{x} = \mathbf{U} g_\theta \mathbf{U}^T \mathbf{x} \quad (12)$$

大多数谱方法都基于公式(12)定义的过程, 而不同方法的区别在于不同过滤器 g_θ 的设计. 最早利用公式(12)进行卷积操作的方法是 Spectral CNN^[47], 它定义过滤器 $g_\theta = \Theta_{i,j}^{(l)}$, 其中, $\Theta_{i,j}^{(l)}$ 是可学习的参数对角阵. Spectral CNN 的图卷积层定义如下:

$$\mathbf{H}_{:,j}^{(l)} = \sigma \left(\sum_{i=1}^{f_{l-1}} \mathbf{U} \Theta_{i,j}^{(l)} \mathbf{U}^T \mathbf{H}_{:,i}^{(l-1)} \right), j = (1, 2, \dots, f_l) \quad (13)$$

其中, l 表示层数, $\mathbf{H}_{:,j}^{(l)}$ 表示节点状态向量第 j 维的值, $\mathbf{H}_{:,j}^{(0)} = \mathbf{X}_{:,j}$, f_{l-1} 表示第 $l-1$ 层通道数, f_l 表示第 l 层通道数. 然而, Spectral CNN 有 3 个较大的缺陷: (1) 由于需要对拉普拉斯矩阵做出矩阵分解, Spectral CNN 最高需要 $O(N^3)$ 的时间复杂度, 这使其无法适用于大型的图结构; (2) 除此之外, 过滤器的学习依赖于特征矩阵 \mathbf{U} , 因此, 当输入不止一个图结构时, 学习的参数无法共享; (3) 在谱域中定义的过滤器并不能保证抓取的是节点领域内的局部信息. 为了解决这些缺陷, ChebNet^[8]和 GCN^[9]相继被提出.

ChebNet 引入了切比雪夫多项式来拟合过滤器 g_θ . 具体地, $g_\theta = \sum_{i=0}^K \theta_i T_i(\Lambda')$, 其中, $\Lambda' = 2\Lambda / \lambda_{\max} - \mathbf{I}_n$. 很显然, Λ' 也是对角矩阵且对角线上所有取值在区间 $[-1, 1]$ 内. 切比雪夫多项式通过下式循环定义: $T_i(\mathbf{x}) = 2\mathbf{x}T_{i-1}(\mathbf{x}) - T_{i-2}(\mathbf{x})$, 其中, $T_0(\mathbf{x}) = 1$, $T_1(\mathbf{x}) = \mathbf{x}$. 从而图卷积层被定义为

$$g_\theta * \mathbf{x} = \mathbf{U} \left(\sum_{i=0}^K \theta_i T_i(\Lambda') \right) \mathbf{U}^T \mathbf{x} \quad (14)$$

通过切比雪夫多项式的拟合, 过滤器能够保证每个节点只受局部信息影响. 这是因为每个多项式过滤器被限制在 K -hop 内, 即在每个卷积层内, 每个节点的表示向量只受到它的 K 阶邻居的影响. 同时, 复杂度被降低到了 $O(KM)$, 其中, K 是多项式项数, M 是边的数量. 显然, 当 K 是一个较小值时, 模型时间复杂度与图中边的数量呈线性关系.

Kipf 和 Welling 等人提出的 GCN^[9]进一步简化了 ChebNet^[8]的过滤器. 他们只使用了切比雪夫多项式的一阶拟合, 即 $K=1$, 并假定 $\lambda_{\max}=2$. 公式(14)被简化为

$$g_\theta * \mathbf{x} = \theta_0 \mathbf{x} - \theta_1 \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{x} \quad (15)$$

GCN 进一步假设 $\theta = \theta_0 = -\theta_1$, 并且为了学习高阶邻居的信息, 模型被扩展成多层:

$$\mathbf{H}^{(l)} = f(\tilde{\mathbf{A}} \mathbf{H}^{(l-1)} \Theta) \quad (16)$$

其中, $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}} \hat{\mathbf{A}} \mathbf{D}^{-\frac{1}{2}}$, 而 $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$, $f(\cdot)$ 表示激活函数, $\mathbf{H}^{(l)}$ 是第 l 层节点状态矩阵, $\mathbf{H}^{(0)} = \mathbf{X}$. 通过叠加卷积层数, 每个节点表示可以融合更加高阶的邻居信息. 然而, Kipf 等人^[9]的实验显示, 当卷积层数超过 2 层后, 叠加更多的层数对模型性能并没有显著提升, 且随着层数的持续增加, 节点特征可能被过平滑^[9], 从而导致模型性能下降. 由于这个原因, 后续的谱方法的卷积层数基本都设置为 2, 而为了融合更加高阶的邻居信息, 人们引入了超图, 这将在第 7 节给出详细叙述.

除了上述介绍的谱方法, CayleyNet^[48]使用 Cayley 多项式替换了切比雪夫多项式, AGCN^[49]通过节点特征

向量间的距离定义了一种残差邻接矩阵, GWNN^[7]引入小波变换替代傅里叶变换.

4.3.2 空间方法

谱方法虽然在图卷积的研究上取得了突破性进展, 具有强大的性能, 但仍存在两项缺陷: (1) 谱方法在推导过程中存在诸多数学假设, 例如限定图结构为无向图, 或假设所有节点特征位于同一特征空间, 这使得谱方法应用的图结构比较有限, 而迁移到更复杂的图结构则需要对谱方法从底层重新推导; (2) 在谱方法中, 图卷积操作需要一次性将整个邻接矩阵输入, 这在图很大时会消耗巨大的内存空间. 同时, 由于模型是直推式的, 如果图中节点或边的关系发生变化, 则需要将整个图输入重新进行训练, 这在需要高时效的应用场景下是无法容忍的.

空间方法的出现有效弥补了谱方法的这些缺陷. 不同于谱方法, 空间方法通过对中心节点的邻居信息进行聚合来更新中心节点的特征表示. 大多数空间方法是归纳式学习, 即模型训练过程中, 只有训练集中的节点结构和特征信息会参与参数更新, 学习得到的映射模式用于测试集的预测. 这使得图结构不必一次性全部输入, 图结构发生变化时也可以只使用变化的点和边重新训练模型.

NN4G^[50]是最早的空间方法之一, 它由多层卷积操作组成, 在每一层, NN4G 通过直接对邻居节点的特征加权求和来实施卷积操作, 通过叠加神经网络层数, 模型可以抓取高阶的邻居信息. 通过引入残差连接和跳跃连接, NN4G 避免了网络层数较深引起的梯度消失. 具体地, NN4G 每一层的更新过程如下:

$$\mathbf{h}_i^{(l)} = \sigma \left(\mathbf{W}_l^T \mathbf{x}_v + \sum_{t=1}^{l-1} \sum_{j \in N(i)} \Theta_l^T \mathbf{h}_j^{t-1} \right) \quad (17)$$

其中, $\sigma(\cdot)$ 表示激活函数, \mathbf{W}_l 和 Θ_l 是每层的系数矩阵, $\mathbf{h}_i^{(0)} = \mathbf{0}$.

另一项对空间方法具有重要意义的研究是 Gilmer 等人提出的 MPNN^[10], 它建立了一个空间方法的统一框架并被后续许多方法沿用. MPNN 认为图卷积可以视为节点之间通过相连边进行信息传递的过程. 它的节点信息更新过程如下:

$$\mathbf{h}_i^{(l)} = \mathcal{F}_1 \left(\mathbf{h}_i^{(l-1)}, \sum_{j \in N(i)} \mathcal{F}_2 \left(\mathbf{h}_i^{(l-1)}, \mathbf{h}_j^{(l-1)}, \mathbf{x}_{ij}^e \right) \right) \quad (18)$$

其中, $\mathcal{F}_1(\cdot)$ 和 $\mathcal{F}_2(\cdot)$ 表示包含可学习参数的函数, $\mathbf{h}_i^0 = \mathbf{x}_i$. 后续研究者通过替换 $\mathcal{F}_1(\cdot)$ 和 $\mathcal{F}_2(\cdot)$ 衍生出了诸多类 MPNN 的方法^[51,52]. 除此之外, Hamilton 等人提出的 GraphSAGE^[11]首次给出了图神经网络的小批量训练方法, 使其在大型网络上具有良好的扩展性. 同时, 考虑到真实网络中 Z 中心节点的邻居数量可能差异很大, GraphSAGE 提出通过采样来获取固定数量的邻居节点. GraphSAGE 的图卷积过程如下:

$$\mathbf{h}_i^{(l)} = \sigma \left(\mathbf{W}^l \cdot \text{CON} \left(\mathbf{h}_i^{(l-1)}, \text{AGG} \left(\left\{ \mathbf{h}_j^{(l-1)}, \forall j \in N^*(i) \right\} \right) \right) \right) \quad (19)$$

其中, $\text{CON}(\cdot)$ 表示连接操作, $\text{AGG}(\cdot)$ 表示聚合函数, 值得注意的是, 不同于之前的图卷积方法, $N^*(i)$ 表示对节点 i 邻居节点的随机采样函数, 返回固定数量的邻居节点. 对于聚合函数的选择, GraphSAGE 提出了均值函数、LSTM 函数和池化函数等多种聚合函数类型. Li 等人^[53]针对各种聚合函数给出了一个统一框架, 他们还发现, 通过残差连接和对卷积层正则化可以帮助训练更深的卷积图神经网络.

Velickovic 等人^[54]则进一步考虑了聚合过程中不同邻居节点对中心节点特征的贡献的差异性. 为了刻画这种差异性, 他们引入了自注意力机制^[55], 并提出了图注意力网络(GAT). 具体地, GAT 通过注意力机制学习每个邻居节点在聚合过程的权重系数, 以此衡量每个节点的重要程度. GAT 的卷积层如下:

$$\mathbf{h}_i^{(l)} = \sigma \left(\sum_{j \in N(i)} \alpha_{ij}^{(l)} \mathbf{W}^{(l)} \mathbf{h}_j^{(l-1)} \right) \quad (20)$$

尽管 MPNN 及其衍生方法取得了优秀的结果, Xu 等人^[56]指出这些方法生成的表示向量对图的信息表示并不完整. 他们提出的图同构神经网络(GIN)对中心节点增加了一个可学习的系数, 并将中心节点和邻居节点的聚合结果输入到一个多层感知机^[56].

值得注意的是, 虽然谱方法和空间方法的出发点和进行图卷积的方式看起来完全不同, 但二者在数学上

具有等价性^[11]. 由于 ChebNet^[8]和 GCN^[9]使用多项式函数或其一阶近似作为过滤器, 它们的卷积操作可以视为同时对所有节点的一阶邻居通过均值函数进行聚合, 因此一些研究者也将它们视为空间方法^[11].

5 异质图神经网络

在真实世界中, 网络中节点和边往往具有多种类型, 而神经网络的早期研究多基于同质图. 由于异质图中复杂的语义关系和多样的特征类型, 传统的基于同质图的 GNN 无法直接迁移到这类图结构. 近年来, 随着神经网络在同质图上的研究逐渐成熟, 研究者们开始将目光投向异质图, 并提出了一系列方法. 我们按照节点特征聚合过程中信息的传递方式将这些方法分成关系类型感知和元路径感知这两种类型. 值得注意的是, 知识图谱由于复杂的语义关系和庞大的规模, 是一种较为特殊的异质图, 而针对它设计的神经网络方法大多是与领域问题相结合^[57,58], 学习图谱中的语义信息辅助进行推断或辨识. 这也与传统异质图 GNN 方法进行节点分类或链接预测的任务不同. 因此, 在上述分类的基础上, 我们将分开讨论普通异质图与知识图谱.

5.1 关系类型感知

5.1.1 普通异质图

异质图与同质图的一个显著区别是异质图中存在多种类型的边, 从而定义了节点间的各种语义关系. 而在神经网络的卷积过程中, 节点间的信息又是通过相连的边进行传递的, 因此一种直接的思路便是按照关系的类型分别设计卷积操作再进行合并.

(1) 非加权方法

最早的关系类型感知方法是 R-GCN^[12]. 与 GCN^[9]和 GraphSage^[11]类似, R-GCN 使用节点特征向量 \mathbf{x}_i 作为输入, 在每个图卷积层中对邻居节点向量进行聚合, 作为中心节点表示向量的更新:

$$\mathbf{h}_i^{(l+1)} = \sigma \left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_i^r} \frac{1}{c_{i,r}} \mathbf{W}_r^{(l)} \mathbf{h}_j^{(l)} + \mathbf{W}_0^{(l)} \mathbf{h}_i^{(l)} \right) \quad (21)$$

其中, \mathcal{N}_i^r 表示节点 i 在关系 r 限制下的邻域, $c_{i,r}$ 是归一化常数, 可以通过训练得到或直接设为 $|\mathcal{N}_i^r|$. 从公式(21)可以看出, 与同质图神经网络不同, R-GCN^[12]为不同类型的关系 r 所连接的节点设置了独立的参数矩阵 $\mathbf{W}_r^{(l)}$.

图 3 表示了 R-GCN 信息的聚合过程. 可以预见, 由于给每种关系分配一个参数矩阵 $\mathbf{W}_r^{(l)}$, 随着异质图中关系类型的增加, 模型可训练参数的数量会迅速增加, 形成过高的模型复杂度, 同时也增加了过拟合的风险. 为了解决这个问题, R-GCN^[12]对参数矩阵 $\mathbf{W}_r^{(l)}$ 做出了一些限制, 从而减少其独立参数的数量. 一种做法是使用一组基矩阵 $\{\mathbf{E}_{1,r}^{(l)}, \dots, \mathbf{E}_{B,r}^{(l)}\}$ 的线性组合作为 $\mathbf{W}_r^{(l)}$, 即 $\mathbf{W}_r^{(l)} = \sum_{b=1}^B \xi_{rb}^{(l)} \mathbf{E}_b^{(l)}$, 其中, $\mathbf{E}_b^{(l)} \in \mathbb{R}^{d^{(l+1)} \times d^{(l)}}$ 是一组基矩阵, 组合系数 $\xi_{rb}^{(l)}$ 与关系 r 有关; 或者将 $\mathbf{W}_r^{(l)}$ 定义为一组块对角阵的直和: $\mathbf{W}_r^{(l)} = \bigoplus_{b=1}^B \mathbf{Q}_{br}^{(l)}$, 其中, $\mathbf{Q}_{br}^{(l)} \in \mathbb{R}^{(d^{(l+1)}/B) \times (d^{(l)}/B)}$ 是块对角矩阵. 与 GraphSAGE^[11]类似, R-GCN^[12]同样定义了半监督和无监督两种学习方式, 分别对应节点分类和链接预测任务. 其中, 节点分类任务使用交叉熵作为损失函数, 链接预测任务借用 DistMult^[59]中的双线性模型对知识图谱中的三元组进行编码, 并引入负采样机制训练模型. 具体的损失函数如下:

$$L = -\frac{1}{(1+\omega)|\mathcal{E}|} \sum_{(i,r,j,\theta) \in \Gamma} \theta \log \sigma(e_i^T R_r e_j) + (1-\theta) \log(1 - e_i^T R_r e_j) \quad (22)$$

(2) 注意力加权方法

虽然 R-GCN^[12]分配了独立的卷积核对应不同类型的节点, 但各类型的节点特征是直接相加的. Zheng 等人^[60]进一步认为邻居节点在通过不同类型的边进行聚合时应该具有不同的权重. 他们提出的 HEP^[60]继承了关系类型感知的思路, 即先是聚合具有相同类型的邻居节点特征, 再对不同种类节点的聚合结果进行连接操作, 以作为中心节点的表示向量. 但与 R-GCN 不同的是, HEP 对异质图中的边也定义了特征向量 $\mathbf{f}(i,j)$, 其中,

i 和 j 表示边 $l(i,j)$ 所连接的两个节点, 并通过 $\mathbf{f}_l(i,j)$ 计算不同邻居节点在聚合过程中的权重 ω_{ij} . 具体地:

$$\omega_{ij} = \sigma(\lambda_{\psi(i,j)} \cdot \mathbf{f}_l(i,j) + b_{\psi(i,j)}) \quad (23)$$

$$\alpha_{ij} = \frac{\omega_{ij}}{\sum_{k \in \mathcal{N}_i^{(c)}} \omega_{ik}} \quad (24)$$

其中, $\Psi(\cdot)$ 表示节点对 (i,j) 到边的类型 $c_l \in C = \{c_1, \dots, c_n\}$ 的映射, $\lambda_{\psi(i,j)}$ 和 $b_{\psi(i,j)}$ 是可训练的参数, $\sigma(\cdot)$ 是激活函数. 由于参数 $\lambda_{\psi(i,j)}$ 和 $b_{\psi(i,j)}$ 与边的类型相关, 因此相同类型的边特征向量共享同一对 $\lambda_{\psi(i,j)}$ 和 $b_{\psi(i,j)}$. 接着, HEP 将邻居节点的特征按照相连边的权重进行聚合:

$$\mathbf{z}_i^c = \sum_{j \in \mathcal{N}_i^{(c)}} \alpha_{ij} \mathbf{h}_j \quad (25)$$

其中, \mathbf{h}_i 表示邻居节点特征向量. 在得到所有类型的边的特征向量 $\{\mathbf{z}_i^{c_1}, \dots, \mathbf{z}_i^{c_n}\}$ 后, HEP 将它们连接在一起并通过一个全连层得到中心节点的特征表示 \mathbf{h}'_i :

$$\mathbf{h}'_i = \sigma(\mathbf{W}' \cdot \mathcal{F}_{concat}(\mathbf{z}_i^{c_1}, \dots, \mathbf{z}_i^{c_n}) + \mathbf{b}') \quad (26)$$

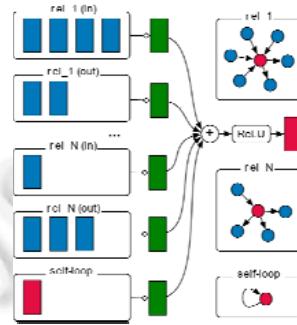


图 3 通过关系类型进行邻居信息聚合的框架图^[12]

HetSANN^[61]也考虑到了不同类型的节点特征在聚合过程中的权重问题, 与 HEP^[60]不同, 它引入了自注意力机制^[54]衡量不同邻居节点的重要程度:

$$\omega_{ij} = \sigma\left(\left[\mathbf{h}_{\phi(i),i}^T \parallel \mathbf{h}_{\phi(i),j}^T\right] \mathbf{a}_r\right), \alpha_{ij} = \frac{\omega_{ij}}{\sum_{k \in \mathcal{N}_i^{(c)}} \omega_{ik}} \quad (27)$$

其中, $\mathbf{h}_{\phi(i),j}^T$ 表示经过线性映射 $\mathbf{h}_{\phi(i),j} = \mathbf{W}_{\phi(i),\phi(j)} \mathbf{h}_j$ 后的节点特征向量, \mathbf{a}_r 表示可学习的注意力参数, 它与边类型 r 相关, \parallel 表示连接操作. 很明显, 公式(23)与公式(27)非常相似, 区别在于 HEP 使用了边的特征计算了权重 ω_{ij} , 而 HetSANN 则使用了节点特征的连接向量进行注意力计算. 为了使结果更加稳定平滑, HetSANN 使用了多头注意力机制并将所有结果进行连接:

$$\mathbf{h}'_i = \parallel_{k=1}^K \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \mathbf{h}_{\phi(i),j} \quad (28)$$

另外, HetSANN 还考虑了异质图中互逆关系之间的语义关联关系. 当给定关系 r 时, 它定义了节点 i 和 j 之间的关系 $i \xrightarrow{r} j$, 显然, 当已知 $i \xrightarrow{r} j$, 可以推知 $j \xrightarrow{\tilde{r}} i$, 其中, \tilde{r} 是 r 的互逆关系. 这反映在自注意力过程中, 即关系 r 和 \tilde{r} 对应的注意力参数 \mathbf{a}_r 和 $\mathbf{a}_{\tilde{r}}$ 并不是相互独立的. 对此, HetSANN 定义 $\mathbf{a}_{\tilde{r}} = -\mathbf{a}_r$, 并将公式(27)改写为

$$\omega_{ij} = \sigma\left(\mathbf{h}_{\phi(i),j}^T (\mathbf{h}_{\phi(i),i} + \mathbf{a}_r)\right) \quad (29)$$

异质图神经网络总结可见表 2.

除了上述介绍的模型, 表 3 也归纳总结了其他关系类型感知方法. 其中, HGT^[13]同样使用了自注意力机制, 它仿照 Transformer^[55]将中心节点的表示定义为查询向量 \mathbf{Q}_i , 将邻居节点表示定义为键值向量 \mathbf{K}_j , 并通过

一个双线性模型 $\mathbf{K}_i \mathbf{W}_{\psi(i,j)}^{\text{ATT}} \mathbf{Q}_j^T / \sqrt{d}$ 计算注意力权重; HetGNN^[62]通过随机游走采样固定数量的邻居节点. 接着, 先对相同类型的节点进行聚合, 再将不同类型的特征合并得到中心节点的表示; ConvE^[63]使用二维卷积和全连层学习节点间的相互作用关系; HGSL^[64]则提出异质图结构具有噪声性和不完整性, 因此在对异质图作表示学习的同时也对图结构进行优化更新. HGA^[65]针对视频问答领域, 将视频中不同帧和问题中不同词组作为节点构造异质图, 并使用多层 GNN 对其进行交叉推理.

表 2 异质图神经网络总结

方法	类型	算法描述	缺陷	应用
HEP ^[60]	关系类型感知 (普通异质图)	使用边的特征计算聚合过程权重	需要额外计算边的特征	社交网络 对齐
HGA ^[65]		多层神经网络进行多模式交叉推理	只适用于视觉问答领域	网络对齐 视觉问答
HetSANN ^[61]		使用 attention 机制衡量异质节点权重	难以迁移到大规模图结构	节点分类 链接预测
HGT ^[13]		引入 attention 机制, 使用双线性模型 计算注意力权重	双线性注意力机制有较高 复杂度	
HetGNN ^[62]		通过重启随机游走得到节点邻居信息, 为不同类型邻域分别实施聚合	难以迁移到大规模图结构	
HGSL ^[64]		学习节点特征的同时更新图结构	难以迁移到大规模图结构	集体分类
HGCN ^[72]		按照边的类型构造子图并分别实施卷积后合并	不是归纳式方法	
KGCN ^[71]	关系类型感知 (知识图谱)	实体与关系的内积作为聚合权重	用户-物品网络未参与训练	推荐系统
KGAT ^[58]		引入 attention 机制, 使用头实体加关系特征与 尾实体的内积作为权重	无法融入实体或关系本身 属性	
KNI ^[57]		知识图谱与用户-物品网络深度融合	无法迁移到其他领域	知识图谱 补充
R-GCN ^[12]		为不同类型关系分配不同卷积核, 通过基矩阵 限制卷积核参数数量	未考虑不同关系间的权重	
ConvE ^[63]		二维卷积层结合全连层	难以捕捉高阶关系	
HAN ^[14]	元路径感知 (普通异质图)	节点级和语义级二级注意力网络	中间节点信息丢失	节点分类 链接预测
MAGNN ^[73]		同时聚合元路径端节点和中间节点信息	元路径编码存在冗余	
GTN ^[74]		根据不同类型边的组合自动构建元路径并通过 多通道 GCN 进行卷积	不是归纳式方法	节点聚类
MCRec ^[75]	元路径感知 (知识图谱)	引入 attention 机制, 通过用户、物品和元路径 特征共同计算权重	需要预先定义元路径	推荐系统
RKGE ^[76]		使用 RNN 实现元路径特征自动编码	基于 RNN 实现导致复杂度 较高	

表 3 动态图神经网络总结

方法	类型	算法描述	缺陷	应用
GaAN ^[83]	基于 RNN (GNN 嵌入)	为多头注意力机制分配权重 并结合 RNN	对长时序数据可能存在梯度消失 或爆炸	交通预测
GCRN ^[84]		ChebNet 结合 LSTM	ChebNet 复杂度较高	图像识别 自然语言处理
RgCNN ^[85]		PATCHY-SAN 结合 LSTM	模型训练过程过于耗时	节点分类
Graph-CNN + LSTM ^[16]		GCN 结合 LSTM	无法应用于大规模网络	交通拥堵预测
DCRNN ^[87]		扩散图卷积网络结合 RNN	可能存在梯度消失或爆炸问题	交通预测
Structural-RNN ^[15]	基于 RNN (RNN 嵌入)	使用 RNN 分别编码节点和边的 特征再通过连接关系聚合	节点和边的语义划分 对结果影响较大	交通预测
EvolveGCN ^[35]		RNN 编码系数矩阵	无法考虑节点的进化关系	链接预测 边分类 节点分类
DynGEM ^[88]	基于自 编码器	动态分配 AE 层数及神经元数量	无法捕捉动态图之间变化关系	链接预测 网络重构
dyngraph2vec ^[90]		将 $\{\mathbf{A}_1, \dots, \mathbf{A}_{t+1}\}$ 合并后输入 AE 得到 \mathbf{A}_{t+1}	模型需要训练的参数过多	链接预测
GRATFEL ^[91]		通过 Graphlet transition 得到 节点对特征并输入 AE	无法融合节点属性特征	

表 3 动态图神经网络总结(续)

方法	类型	算法描述	缺陷	应用
TO-GAE ^[92]	基于自编码器	两层 GCN 作为编码器	对图结构的更新带来训练复杂度的增加	链接预测
DKGE ^[93]		使用 Attentive GCN 对变化子图结构进行增量学习	需要同时训练节点子图和关系子图	知识图谱补全 视觉问答
ST-GCN ^[18]	时空图神经网络	仿照 CNN 定义时空卷积	只适用于同质图	动作识别
Stacked-STGCN ^[94]		在 ST-GCN 基础上引入堆叠沙漏结构	适合更大范围的时空图但相应复杂度更高	动作分割
STGCN ^[95]		固定卷积核进行时空卷积操作	较高复杂度使得应用于大规模数据有困难	交通预测
ASTGCN ^[96]		分别使用空间与时间注意力为不同节点在时空维度分配权重		交通预测
PST-GCN ^[97]		自动优化时空图神经网络结构		动作识别

5.1.2 知识图谱

知识图谱(knowledge graph, KG)是另一类较为特殊的异质图, 它利用三元组(头实体、关系和尾实体)作为基本元素描述事物间的事实关系. 相比于社交网络或生物分子网络等异构网络, 知识图谱的节点和关系类型要丰富得多, 图谱规模往往也非常巨大. 这使得知识图谱具有更高的异质性和更复杂的语义关系. 早期对 KG 的研究主要集中在知识图谱补全和事实推断上. 由于需要在大规模图谱上进行应用, 这一时期的方法更加注重算法的简单、高效. 例如, Bordes 等人提出的 TransE^[66]和后续衍生的一系列 Trans 方法(TransH^[67]、TransR^[68]), 通过翻译规则 $\mathbf{e}_h + \mathbf{e}_r \approx \mathbf{e}_t$ 作为三元组关系限制, 学习实体和关系的嵌入表示. DistMult^[59]、RESCAL^[69]和 Simple^[70]等方法则利用双线性函数 $\mathbf{e}_h^T \mathbf{A}_r \mathbf{e}_t$ 编码三元组内实体关系. 这些方法虽然复杂度较低, 但无法捕捉知识图谱更加高阶的语义关系. 由于不涉及图神经网络, 这些方法不在本文讨论范畴.

近年来, 随着图神经网络在简单图和异质图上的成功应用, 研究者开始尝试利用 GNN 对知识图谱进行更深层次的关系挖掘. 与 HetSANN^[61]等方法相似, 这些方法按照关系类型定义不同的卷积核或注意力系数, 因此我们将它们也归类于关系类型感知方法. 其中, Wang 等人^[71]提出一个端到端的知识图谱卷积网络 KGCN, 并将学习到的实体特征用于辅助推荐系统. 他们计算中心实体(即用户)与关系特征的内积, 作为邻居实体聚合过程中的权重 $\pi_{r,v,e}^u$. 通过多次迭代, 可以获取图谱中 K -hop 的邻居信息, 再通过聚合器将邻居特征与中心实体特征融合:

$$\mathbf{e}^{[u,(l+1)]} = \sigma(\mathbf{W} \cdot (\mathbf{e}^{[u,(l)]} + \mathbf{e}_{S(v)}^u) + \mathbf{b}) \tag{30}$$

其中, $S(v)$ 表示中心实体 v 的 K 阶邻居 $\mathbf{e}_{S(v)}^u$, 通过 $\pi_{r,v,e}^u$ 与邻居特征 \mathbf{e} 加权求和得到. 最终得到的实体特征 $\mathbf{e}^{[u,(L)]}$ 与用户特征 \mathbf{u} 通过映射函数 $\hat{y} = f(\mathbf{e}^{[u,(L)]}, \mathbf{u})$ 计算喜好概率 \hat{y} . 图 4 展示了 KGCN^[71]在 $K=2$ 情况下的聚合过程及概率预测.

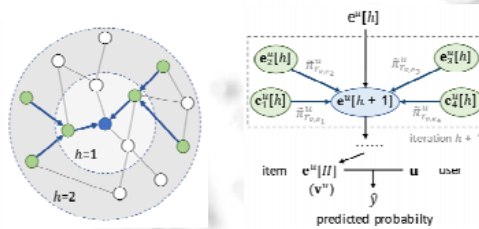


图 4 KGCN^[71]模型图

除此之外, KGAT^[58]进一步将知识图谱与用户-物品二分图融合成一个统一的图结构, 并使用图注意力网络^[54]对融合图进行学习. KNI^[57]将用户-物品关系与知识图谱进行更加复杂的融合.

5.2 元路径感知

关系类型感知方法按照直接相连边的类型分别进行特征聚合, 这样虽然考虑了不同关系类型带来的异质性, 但存在两大缺陷: (1) 关系类型感知方法的每个卷积层只能考虑一阶邻居节点的信息, 然而在异质图中, 不同关系的进一步组合将形成更加高阶的语义关联, 而这些复杂语义关联是关系类型感知方法难以捕捉的; (2) 随着关系类型的增加, 模型可训练参数将迅速增加, 使得模型难以应用到大规模网络当中。

近年来, 越来越多的研究者开始将元路径(meta-path)融入到异质图表示学习中, 从而简化图结构以及获取更加高阶的关联关系. 元路径是指异质图中描述了两个节点间复合关系的路径信息. 元路径可以定义为任意关系的任意长度的组合, 因此可以描述更加复杂的语义关联. 例如图 5 所示的元路径描述了在同一会议上发表了文章的作者之间的关联.

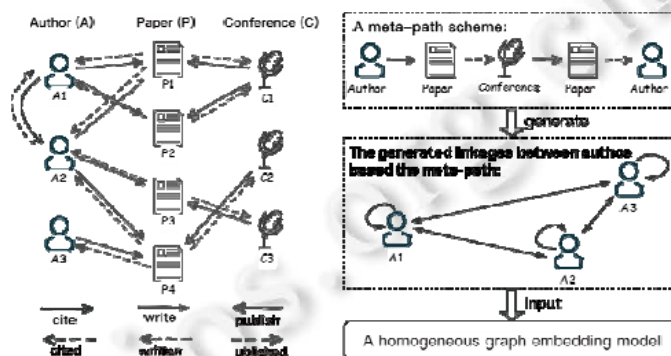


图 5 元路径以及通过元路径构造的同质图^[61]

Dong 等人^[77]最早利用元路径来获取异质图中复杂的语义关系. 他们提出的 Metapath2vec^[77]延续了 DeepWalk^[1]和 Node2vec^[2]中随机游走的思路, 通过元路径限制随机游走的路径, 从而获取节点周围的邻居信息. HIN2Vec^[78]进一步为节点对间出现的元路径分配了概率值, 后续的 HHNE^[79]、GHE^[80]等方法则对 Metapath2vec^[77]进行了各种改进. 然而这些方法也继承了传统网络表示学习方法的缺陷, 如无法融合节点本身的属性特征, 或只考虑到某一种元路径而忽略了其他类型.

5.2.1 普通异质图

不同于上述基于元路径的方法, HAN^[14]引入了图注意力网络对构造的同质图进行学习. 如图 2 所示, 它先通过元路径获取异质图中的节点对, 再利用它们构造出基于元路径的同质图, 进而可以使用传统的 GNN 方法对这些同质图进行表示学习. 具体地, 对于某一节点 i , \mathcal{N}_i^ξ 表示通过元路径 ξ 与节点 i 相连的节点所构成的邻域. 获取同一类型的所有节点 $i \in \mathcal{V}^C$ 的邻域可以得到基于元路径的同质图 \mathcal{G}^ξ , 而通过定义不同的元路径 ξ_1, \dots, ξ_r , 可以得到一系列基于元路径的同质图 $\mathcal{G}^{\xi_1}, \dots, \mathcal{G}^{\xi_r}$.

为了有效学习这些同质图, HAN^[14]设计了一个两级图注意力网络. 具体地, 首先对同质图 \mathcal{G}^ξ 内的节点特征进行聚合, 它引入节点级注意力机制为不同节点分配不同的权重:

$$\alpha_{ij}^\xi = \frac{\exp(\sigma(\mathbf{a}_\xi^T \cdot [\mathbf{h}'_i \parallel \mathbf{h}'_j]))}{\sum_{k \in \mathcal{N}_i^\xi} \exp(\sigma(\mathbf{a}_\xi^T \cdot [\mathbf{h}'_i \parallel \mathbf{h}'_k]))} \quad (31)$$

$$\mathbf{z}_i^\xi = \parallel \sigma \left(\sum_{j \in \mathcal{N}_i^\xi} \alpha_{ij}^\xi \mathbf{h}'_j \right) \quad (32)$$

其中, \mathbf{h}'_i 表示经过线性映射后的节点特征, \mathbf{a}_ξ 表示与元路径相关的注意力向量, \mathbf{z}_i^ξ 是节点 i 基于元路径 ξ 的节点特征. 通过对不同的同质图 $\mathcal{G}^{\xi_1}, \dots, \mathcal{G}^{\xi_r}$ 分别进行特征聚合, 可以得到相应的节点特征 $\{\mathbf{Z}^{\xi_1}, \dots, \mathbf{Z}^{\xi_r}\}$, 为了衡量不同元路径特征的重要性, HAN^[14]引入语义级注意力机制:

$$\beta_{\xi} = \frac{\exp\left(\frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{q}^T \cdot \tanh(\mathbf{W}_{\xi} \cdot \mathbf{z}_i^{\xi} + \mathbf{b})\right)}{\sum_{\xi=1}^C \exp\left(\frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \mathbf{q}^T \cdot \tanh(\mathbf{W}_{\xi} \cdot \mathbf{z}_i^{\xi} + \mathbf{b})\right)} \quad (33)$$

$$Z = \sum_{\xi \in \Phi_{\xi}} \beta_{\xi} \cdot \mathbf{Z}_{\xi} \quad (34)$$

其中, C 是元路径种类数, \mathbf{q} 是语义注意力向量. 对比公式(21)和公式(32)可以看出, 基于元路径的同质图可以将原异质图中高阶关联的节点对转换成紧密相连的一阶邻居, 这缩短了节点间信息传递需要的跳数; 而通过变更元路径, 又可以得到具有不同语义信息的全局图, 使得 HAN 学习到更加全面的节点表示.

然而, HAN 也有着明显的缺陷. HAN 只保留了元路径两端的节点以构造同质图, 而丢弃了元路径的中间信息, 这不可避免地造成了信息损失; 同时, 这也使得模型无法同时得到异质图中所有类型节点的特征表示. MAGNN^[73] 率先尝试解决这个问题, 它在聚合过程中同时考虑了基于元路径的邻居节点信息和元路径上的中间节点信息. 为了实现这一目的, MAGNN^[73] 提出一个元路径实例编码器, 它将元路径上所有节点特征作为输入, 将它们编码成单个特征向量作为输出. 具体地:

$$\mathbf{h}_{ij}^{\xi} = f_{MIE}\left(\left\{\mathbf{h}_k \mid k \in \mathcal{P}_{i \rightarrow j}^{\xi}\right\}\right) \quad (35)$$

其中, $\mathcal{P}_{i \rightarrow j}^{\xi}$ 表示元路径实例, $f_{MIE}(\cdot)$ 表示元路径实例编码器, 能够将元路径实例上沿 i 到 j 的所有节点信息编码成单个特征向量 \mathbf{h}_{ij}^{ξ} . 在编码了所有元路径实例后, 与 HAN 相似, MAGNN^[73] 设计了两层注意力机制. 首先使用元路径内部注意力机制计算不同元路径实例特征 \mathbf{h}_{ij}^{ξ} 在聚合过程中的权重:

$$\alpha_{ij}^{\xi} = \frac{\exp(\sigma_L(\mathbf{a}_{\xi}^T [\mathbf{h}_i^{\xi} \parallel \mathbf{h}_{ij}^{\xi}]))}{\sum_{k \in \mathcal{N}(i)} \exp(\sigma_L(\mathbf{a}_{\xi}^T [\mathbf{h}_k^{\xi} \parallel \mathbf{h}_{ik}^{\xi}]))} \quad (36)$$

$$\mathbf{h}_i^{\xi} = \sigma\left(\sum_{i \in \mathcal{N}_{\xi}(i)} \alpha_{ij}^{\xi} \mathbf{h}_{ij}^{\xi}\right) \quad (37)$$

其中, \mathbf{a}_{ξ} 表示注意力向量, $\sigma_L(\cdot)$ 表示 *LeakyReLU* 激活函数, \mathbf{h}_i^{ξ} 是节点 i 在元路径 ξ 下的聚合的特征向量, 为了衡量不同元路径下的节点特征的重要性, MAGNN^[73] 设计了元路径外部注意力机制:

$$\left. \begin{aligned} e_{\xi} &= \mathbf{q}_A^T \cdot \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \tanh(\mathbf{W}_A \cdot \mathbf{h}_v^{\xi} + \mathbf{b}_A), \\ \beta_{\xi} &= \frac{\exp(e_{\xi})}{\sum_{\xi=1}^C \exp(e_{\xi})}, \\ \mathbf{h}_i &= \sigma\left(\mathbf{W}_o \left(\sum_{\xi \in \Phi_{\xi}} \beta_{\xi} \cdot \mathbf{h}_i^{\xi}\right)\right) \end{aligned} \right\} \quad (38)$$

其中, Φ_{ξ} 表示元路径种类集合, \mathbf{h}_i 是最终输出的节点特征表示. HAN^[14] 和 MAGNN^[73] 虽然利用元路径实现了异质图中高阶关联的信息聚合, 但二者的元路径均需事先定义, 而筛选合理的元路径要求研究者需要一定的领域知识.

为了弥补这个缺陷, Yun 等人提出了 GTN^[74] 来根据异质图边的类型自动构建出有意义的多跳连接, 即元路径. GTN 将异质图表示为一系列邻接矩阵的集合 $\{\mathbf{A}_k\}_{k=1}^K$, 其中, $K = |\mathcal{B}_e|$ 表示边类型数量. Yun 等人^[74] 指出, 元路径连接的节点的邻接矩阵等于元路径上边的邻接矩阵的内积, 即 $\mathbf{A}_p = \mathbf{A}_{l_1} \dots \mathbf{A}_{l_2} \mathbf{A}_{l_1}$. 给定元路径长度为 l 时, GTN 将所有邻接矩阵的线性组合作为第 l_i 条边对应的邻接矩阵, 则元路径邻接矩阵 \mathbf{A}_p 可表示为

$$\mathbf{A}_p = \left(\sum_{t_1 \in \mathcal{B}_e} \alpha_{t_1}^{(1)} \mathbf{A}_{t_1} \right) \left(\sum_{t_2 \in \mathcal{B}_e} \alpha_{t_2}^{(2)} \mathbf{A}_{t_2} \right) \dots \left(\sum_{t_l \in \mathcal{B}_e} \alpha_{t_l}^{(l)} \mathbf{A}_{t_l} \right) \quad (39)$$

其中, $\alpha_{t_l}^{(l)}$ 表示相应系数. 这样一来, 只需给定元路径长度 l , GTN 可以自动生成对应的元路径邻接矩阵, 从而避免了借助领域知识人为定义元路径. 由于异质图往往存在多种元路径, GTN 设计了一个多通道卷积神经网络以便同时处理不同元路径生成的图结构, 再将卷积结果相连接得到最终节点特征表示:

$$\mathbf{Z} = \left\| \sigma \left(\tilde{\mathbf{D}}_i^{-1} \tilde{\mathbf{A}}_p \mathbf{X} \mathbf{W} \right) \right\|_{i=1}^C \quad (40)$$

其中, $\tilde{\mathbf{A}}_p$ 表示加入自连接后的元路径邻接矩阵, $\tilde{\mathbf{D}}_i$ 是 $\tilde{\mathbf{A}}_p$ 的度矩阵. 虽然 GTN 无需领域知识, 但在每次生成 \mathbf{A}_p 时需要对各条边对应的邻接矩阵进行一维卷积, 而在 HAN 中, \mathbf{A}_p 可以预先定义好. 因此相比于 HAN^[14], GTN 拥有更高的复杂度, 而这在应用到大规模图结构时是不利的.

5.2.2 知识图谱

基于元路径感知的方法同时也在知识图谱学习中得到广泛使用. 以推荐系统为例, 在进行物品推荐时, 研究者往往会构建用户-物品间的关联图, 借助网络中物品或用户的结构相似性或关联关系进行推荐. 近年来, 研究者进一步引入知识图谱, 将其与用户-物品关联图相融合, 构造更大、更全的关联网络, 并相应提出一系列基于知识图谱的推荐系统方法^[75,81]. 其中, 元路径起到了重要作用.

在用户-物品关联网络中, 特定用户与物品间通过一系列节点与关系组合相连, 亦即元路径. 一方面, 不同元路径反映了用户物品间的不同语义关系, 因而对于推荐具有不同的重要性, 另一方面, 元路径清晰的语义关系可以为推荐提供良好的可解释性. 由于元路径在知识图谱推荐系统中的重要作用, Yu 等人^[81]很早便使用多种元路径计算物品间的相似度, 后续的 HeteRec^[81]进一步使用元路径信息计算用户与物品间的关联关系, HeteRec-p^[82]指出不同元路径对用户的不同重要性, 并通过预先对用户聚类为不同类用户分配不同权重.

近年来, 随着图注意力网络(GAT^[54])的提出, Hu 等人提出的 MCRec^[75]进一步引入自注意力机制衡量不同元路径的重要性 $\alpha_{u,i,\rho}$, 并使用学习得到的权重用于更新元路径特征表示 $\mathbf{c}_{u \rightarrow i}$ 和用户与物品嵌入向量 $\tilde{\mathbf{x}}_u$ 和 $\tilde{\mathbf{y}}_i$. 用户和物品间可通过不同元路径相连, 其中某条元路径 ρ 的权重 $\alpha_{u,i,\rho}$ 通过用户、元路径和物品特征共同定义:

$$\alpha_{u,i,\rho}^{(1)} = f \left(\mathbf{W}_u^{(1)} \mathbf{x}_u + \mathbf{W}_i^{(1)} \mathbf{y}_i + \mathbf{W}_\rho^{(1)} \mathbf{c}_\rho + \mathbf{b}^{(1)} \right) \quad (41)$$

$$\alpha_{u,i,\rho}^{(2)} = f \left(\mathbf{w}^{(2)\top} \alpha_{u,i,\rho}^{(1)} + \mathbf{b}^{(2)} \right) \quad (42)$$

其中, $f(\cdot)$ 表示 ReLU 激活函数. 通过对不同元路径特征加权求和得到 u 和 i 之间的综合元路径表示 $\mathbf{c}_{u \rightarrow i} = \sum_{\rho \in \mathcal{M}_{u \rightarrow i}} \alpha_{u,i,\rho} \cdot \mathbf{c}_\rho$. 特定元路径与用户或物品相互作用会产生特定的用户和物品特征表示, 因此, MCRec 将用户特征和物品特征分别与元路径特征相融合得到更新后的 $\tilde{\mathbf{x}}_u$ 和 $\tilde{\mathbf{y}}_i$, 并计算用户与物品间的推荐概率:

$$\hat{r}_{u,i} = MLP(\tilde{\mathbf{x}}_u \oplus \mathbf{c}_{u \rightarrow i} \oplus \tilde{\mathbf{y}}_i) \quad (43)$$

其中, \oplus 表示连接操作. 除此之外, 后续的 RKGE^[76]在 MCRec^[75]的基础上进一步做出改进, 通过引入循环神经网络自动编码用户-物品间的元路径特征, 而不需要预先定义元路径.

异质图神经网络成功地将图神经网络扩展到边和节点异质的图结构, 在不同领域取得了优秀的结果. 关于普通异质图与知识图谱的研究仍是相对独立的两个方向. 因此, 如何将已在普通异质图上已经成熟的图神经网络方法迁移到节点和边的类型丰富得多的知识图谱领域仍是当前异质网络的研究热点之一.

6 动态图神经网络

真实世界中的图往往具有时序性, 即网络结构和节点属性会随着时间动态变化. 传统图神经网络^[9]基于静态图结构, 因此无法直接用于处理动态图. 为了解决这个问题, 研究者们提出了一系列基于时序图的图神经网络. 这些方法按照处理时序信息的方式可以分为基于循环神经网络、基于自编码器和时空图神经网络这 3 种类型. 所有动态图 GNN 的分类及描述见前文表 3.

6.1 基于循环神经网络

动态图的特征和结构信息随时间发生变化, 它的节点属性矩阵 $(\mathbf{X}_1, \dots, \mathbf{X}_t)$ 以及邻接矩阵 $(\mathbf{A}_1, \dots, \mathbf{A}_t)$ 可以看作一组序列数据. 而循环神经网络(recurrent neural network, RNN)等序列模型在处理序列数据上有着很好的效果. 因此一个直接的想法是, 将 GNN 与诸如 RNN 等序列模型相结合处理动态网络. 按照是将 GNN 作为循环单元嵌入 RNN 还是将 RNN 作为参数单元嵌入 GNN, 这类方法可以进一步分为如下两种类型.

6.1.1 GNN 嵌入 RNN

这类方法首先通过卷积层获取各时刻的节点特征, 再输入循环神经网络进行特征融合. 因此, GNN 可以视为嵌入 RNN 的循环单元. 这些方法可以使用以下框架统一表示:

$$\mathbf{Z}^{(t)} = \mathcal{F}_{GNN}(\mathbf{X}^{(t)}, \mathbf{A}; \mathbf{W}) \tag{44}$$

$$\mathbf{H}^{(t)} = \mathcal{F}_{RNN}(\mathbf{H}^{(t-1)}, \mathbf{Z}^{(t)}) \tag{45}$$

其中, $\mathcal{F}_{GNN}(\cdot)$ 表示图卷积层, $\mathbf{X}^{(t)}$ 是 t 时刻的节点特征矩阵, $\mathcal{F}_{RNN}(\cdot)$ 表示序列式模型. 不同序列式方法体现在卷积函数 $\mathcal{F}_{GNN}(\cdot)$ 和序列模型 $\mathcal{F}_{RNN}(\cdot)$ 的不同设计上. Zhang 等人提出的 GaAN^[83]是较早使用上述框架的模型. GaAN 针对图注意力网络 GAT^[54]的多头注意力机制做出改进, 为每一头注意力进一步分配权重:

$$\mathbf{x}_i^{(t)} = \mathbf{x}_i^{(t-1)} \oplus \left\| \left\| \left(\mathbf{g}_i^{(k)} \sum_{j \in \mathcal{N}_i} \omega_{i,j}^{(k)} \mathcal{F}_\theta(\mathbf{x}_j^{(t-1)}) \right) \right\| \right\| \tag{46}$$

其中, $\mathbf{g}_i^{(k)} = \psi_g(\{\mathbf{x}_i; \mathbf{x}_j | j \in \mathcal{N}(i)\})$ 是多头注意力系数, 对应第 k 头注意力的权重, 通过中心节点和邻居节点的特征计算. 此外, Zhang 等人^[83]进一步将 GaAN 与 RNN 相结合用于预测交通速度.

除了 GaAN 之外, 公式(44)和公式(45)还衍生出了一系列方法. Seo 等人^[84]将 ChebNet^[8]作为循环单元嵌入 LSTM; Narayan 等人^[85]将空间方法 PATCHY-SAN^[86]嵌入 LSTM; Mohanty 等人^[16]将 GCN^[9]嵌入 LSTM. Li 等人^[87]对公式(45)的框架稍作改进, 将扩散图卷积网络作为循环单元, 并采用了自编码器结构, 从而可以预测 k 步节点特征.

6.1.2 RNN 嵌入 GNN

不难发现, 如果使用 GNN 嵌入 RNN 的方法, 图卷积层的每次输出都是相互独立的, 即每个时刻 $\mathcal{F}_{GNN}(\cdot)$ 只能使用当前时刻的节点特征进行图卷积, 而无法利用历史节点特征. 而 RNN 嵌入 GNN 的方法与之正好相反, 它先通过 RNN 编码图中节点和边的时序信息, 这样, GNN 就可以使用节点及边的历史信息用于特征聚合. Structural-RNN^[15]就是一种 RNN 嵌入 GNN 的方法. 它对节点和边分别设计了 RNN 以编码图中复杂的时序关联关系. 具体地, Structural-RNN^[15]首先将图中的节点和边划分成不同的语义组, 位于同一语义组中的节点或边将共享相同的 RNN 参数, 这样就不需要对每种节点或边设计一个独立的 RNN; 接着, 模型按照网络中的连接关系将边的编码特征聚合到节点特征上. 图 6 给出了 Structural-RNN^[15]的前向传播过程.

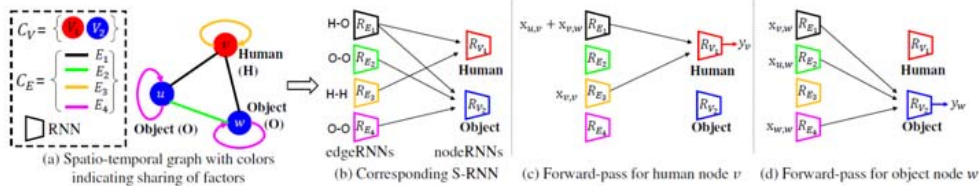


图 6 Structural-RNN^[15]模型框架图

Pareja 等人提出的 EvolveGCN^[35]则将 GCN 的系数矩阵 $\mathbf{W}_{t-1}^{(l)}$ 输入 RNN 进行编码, 从而融入之前时刻的系数矩阵信息, 再输出当前时刻的系数矩阵 $\mathbf{W}_t^{(l)}$. Pareja 等人给出了基于 GRU 和 LSTM 的两种编码方式:

$$\mathbf{H}^{(t+1)} = \sigma(\hat{\mathbf{A}}_t \mathbf{H}_t^{(l)} \mathbf{W}_t^{(l)}) \tag{47}$$

$$\mathbf{W}_t^{(l)} = GRU(\mathbf{H}_t^{(l)}, \mathbf{W}_t^{(l-1)}), \text{ or } \mathbf{W}_t^{(l)} = LSTM(\mathbf{W}_t^{(l-1)}) \tag{48}$$

EvolveGCN^[35]的另一个优势在于对系数矩阵更新的过程可以不需要节点特征, 因此不必将整个时间跨度

上的节点特征输入 RNN, 这有利于处理不同时间戳下节点集合变化较大的网络结构.

6.2 基于自编码器

Goyal 等人提出的 DynGEM^[88]使用与 SDNE^[37]类似的图自编码器学习动态图结构. DynGEM 的基本思路是使用自编码器 $AE^{(t-1)}$ 学习 $t-1$ 时刻的快照 $\mathcal{G}^{(t-1)}$ 并得到对应的节点嵌入向量 $\mathbf{Z}^{(t-1)}$, 为了处理时刻 t 的快照 $\mathcal{G}^{(t)}$, $AE^{(t)}$ 使用 $AE^{(t-1)}$ 的参数进行初始化. 考虑到图结构会随时间变化, DynGEM 根据 $\mathcal{G}^{(t)}$ 与 $\mathcal{G}^{(t-1)}$ 的差异程度为 $AE^{(t)}$ 给定新的层数和每层的神经元数量. 当 $AE^{(t)}$ 相对 $AE^{(t-1)}$ 发生变化时, 该模型使用 Net2WiderNet 方法^[89]以保证自编码器层数和每层神经元数量发生变化时还能大致保留每个全连层之间的映射关系.

DynGEM^[88]没有将 GNN 作为循环单元, 因此具有较少的训练参数和较低的复杂度. 然而它并不能很好地捕捉动态图之间的变化关系. 因此, Goyal 等人进一步提出 $\text{dyngraph2vec}^{[90]}$, 它利用 t 时刻到 $t+l$ 时刻的邻接矩阵 $\{\mathbf{A}_t, \dots, \mathbf{A}_{t+l}\}$ 作为输入, 利用一个自编码器得到 $t+l+1$ 时刻的邻接矩阵 \mathbf{A}_{t+l+1} . Goyal 等人给出了多种编码和解码方式. 如图 7 所示, dyngraph2vecAE 使用连接操作将 $\{\mathbf{A}_t, \dots, \mathbf{A}_{t+l}\}$ 相连并直接输入 AE, dyngraph2vecAERNN 则将 $\{\mathbf{A}_t, \dots, \mathbf{A}_{t+l}\}$ 通过 LSTM 编码再输入全连层得到 \mathbf{A}_{t+l+1} . dyngraphec 两种基于自编码器的模型结构如图 7 所示.

除了上述方法之外, Rahman 等人^[91]将节点对作为自编码器输入. 在每个时间戳, 他们通过 Graphlet transition^[91]的方式计算每个节点对的特征, 再将过去时间戳上所有节点对的特征相连接并输入一个自编码器以学习节点表示. Bonner 等人^[92]基于一个两层 GCN 构成的 AE, 可以基于当前快照的图结构重建下一个快照下的节点邻域. DKGE^[93]通过对动态知识图谱变化的子图结构进行增量学习, 有效降低了动态图谱更新的复杂度.

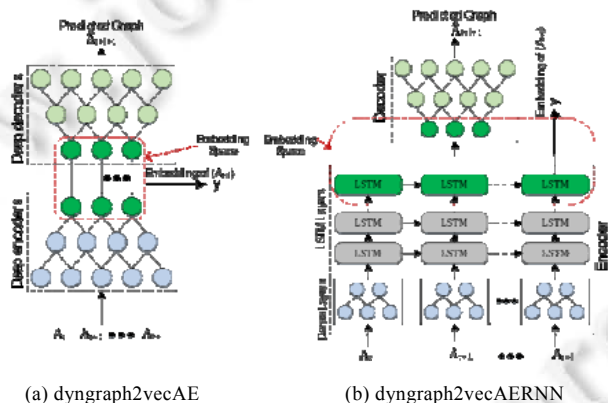


图 7 $\text{dyngraph2vec}^{[90]}$ 模型结构

6.3 时空图神经网络

时空图(spatial-temporal graph)是另一类特殊的动态图. 它通过将不同快照上的相同节点连接, 将动态网络组合成了一个静态整体. 在时空图中, 既有快照内节点间的空间连接, 又有快照之间节点的时间连接. 时空图因此将动态的序列信息转换成了静态的时间连接关系. 相应地, 时空图不再需要结合 RNN, 而是直接使用 GNN 对空间和时间关系进行卷积, 从而避免了 RNN 方法迭代过程中的高复杂度以及梯度消失等问题. 这类方法被统称为时空图神经网络(spatial-temporal graph neural network, STGNN).

Yan 等人的 ST-GCN^[18]首先提出了使用一维卷积核分别对时空图中空间和时间信息进行卷积. ST-GCN 首先仿照 CNN 对于二维图像的卷积方式定义了图上的空间卷积:

$$f_{out}(v_i) = \sum_{v_j \in \mathcal{N}(v_i)} \frac{1}{Z_i(v_i)} f_{in}(\mathbf{p}(v_i, v_j)) \cdot w(v_i, v_j) \quad (49)$$

其中, $\mathcal{N}(v_i) = \{v_j | d(v_j, v_i) \leq K\}$ 表示中心节点的 K 阶邻域, $\mathbf{p}: \mathcal{N}(v_i) \rightarrow \mathcal{V}$ 表示采样函数, 当给定采样范围为

1 阶邻居时, $\mathbf{p}(v_i, v_j) = v_j$. $\mathbf{w}(v_i, v_j)$ 表示权重函数, 对应 CNN 中的卷积核, 为了避免引入过多参数, ST-GCN 将 $\mathcal{N}(v_i)$ 中的节点映射到 K 个子集 $l_i: \mathcal{N}(v_i) \rightarrow \{0, \dots, K-1\}$, 位于相同子集的节点共享相同的权重函数. 而对于时间卷积, 则只需将节点的邻域在时间上进行扩展即可:

$$N(v_i) = \left\{ v_j \mid d(v_j, v_i) \leq K, |q-t| \leq \frac{\Gamma}{2} \right\} \quad (50)$$

空间映射函数 $l_i(\cdot)$ 可以有多种选择, 最简单的方式是单标签, 即给所有节点分配相同权重, 这样一来, 公式(49)退化成与 GCN^[9]相同的卷积方式, 对应的矩阵化表达如下:

$$\mathbf{f}_{out} = \Lambda^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\Lambda^{-\frac{1}{2}}\mathbf{f}_{in}\mathbf{W} \quad (51)$$

在 ST-GCN 的基础上, Ghosh 等人^[94]引入堆叠沙漏结构, 利用编码解码结构使得模型具有更好的泛化性能和定位精度. Guo 等人^[96]则进一步引入注意力机制为不同节点在空间维度和时间维度上分配不同的权重. ASTGCN 通过一个双线性函数定义节点在空间和时间上的相似度. 具体地:

$$\mathbf{S} = \mathbf{V}_s \sigma \left(\left(\mathcal{X}_h^{(r-1)} \mathbf{W}_1 \right) \mathbf{W}_2 \left(\mathbf{W}_3 \mathcal{X}_h^{(r-1)} \right)^T + \mathbf{b}_s \right) \quad (52)$$

$$\mathbf{E} = \mathbf{V}_e \sigma \left(\left(\mathcal{X}_h^{(r-1)} \mathbf{U}_1 \right) \mathbf{U}_2 \left(\mathbf{U}_3 \mathcal{X}_h^{(r-1)} \right)^T + \mathbf{b}_e \right) \quad (53)$$

其中, \mathbf{S} 和 \mathbf{E} 分别是空间相似度和时间相似度矩阵, $\mathcal{X}_h^{(r-1)} = (\mathbf{X}_1, \dots, \mathbf{X}_{T_{r-1}})$ 是初始时刻到 T_{r-1} 时刻节点特征序列, \mathbf{V}_s 、 \mathbf{V}_e 、 \mathbf{W}_1 、 \mathbf{U}_1 、 \mathbf{W}_2 、 \mathbf{U}_2 、 \mathbf{W}_3 、 \mathbf{U}_3 、 \mathbf{b}_s 、 \mathbf{b}_e 是可训练的参数矩阵, 再通过 softmax 函数可以得到节点间的注意力系数:

$$\mathbf{P}'_{i,j} = \frac{\exp(\mathbf{P}_{i,j})}{\sum_{j=1}^N \exp(\mathbf{P}_{i,j})} \quad (54)$$

其中, $\mathbf{P}_{i,j} = \mathbf{S}_{i,j}$ 或 $\mathbf{E}_{i,j}$, 分别对应空间或时间卷积过程中的注意力系数. ASTGCN 的模型框架如图 8 所示.

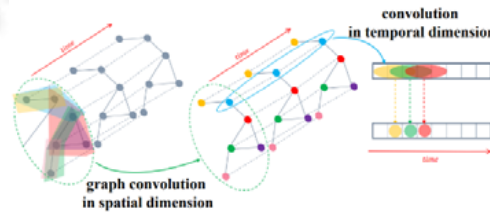


图 8 注意力时空图神经网络 ASTGCN^[96]模型框架

除了上述提到的方法, STGCN^[95]使用了与 Yan 等人^[18]相似的框架, 直接在时空图上进行卷积操作来预测交通流问题. Heidari 等人^[97]考虑到现有时空图神经网络复杂的拓扑结构, 给出了通过渐进方式逐步优化网络深度和宽度的 PST-GCN^[97].

时空图将动态的时序关系转化成了静态的时间连接, 避免了使用序列模型带来的高复杂度与梯度问题, 在交通流预测与动作识别领域已经取得了优秀成果, 是当前动态图神经网络的研究热点之一.

7 超图神经网络

在真实世界中, 事物间的关联关系往往不是成对出现的, 而是两个以上实体间共同构成相互作用关系. 直接使用简单图表示这种非成对关系将会带来信息损失^[19]. 超图(hypergraph)扩展了简单图的定义. 在超图中, 一条超边可以包含任意数量的节点, 因此可以直接储存非成对关系. 超图相比于简单图拥有更加灵活的边定义方式, 因而对于复杂关系有着更加强大的表达能力. 对于超图的研究最早可以追溯到 20 世纪 80 年代, 这一时期的研究集中在对超图数学性质的推导上^[98-100]. 更近些时候, 伴随着机器学习方法在图结构上的成功应用, 越来越多的研究者再次将目光投向超图, 并借助机器学习方法对超图结构进行更深层次的挖掘.

由于超图对于多元关系的建模能力, 早期对于超图的研究主要集中于领域化的问题. 其中, Tan 等人^[101]率先利用超图结构进行了社交网络中用户对齐的研究; Zhu 等人^[102]和 Bu 等人^[103]则关注超图在推荐系统中的应用; Fatemi 等人^[104]将基于三元组的传统知识图谱结构扩展到了知识超图形式; Hwang 等人^[105]和 Klamt 等人^[106]利用超图进行蛋白质或细胞网络的多元关系建模; Huang 等人^[107]则将超图应用到计算机视觉上. 超图的应用将在第 8.3 节详细介绍. 尽管这一时期的研究者已经开始利用超图对各领域中的多元关系进行建模, 但这些方法仍然缺乏对超图表示学习系统的研究, 因此不具有普适性, 很难迁移到其他领域.

直到近年来, 随着图神经网络在简单图结构上的逐渐成熟, 研究者开始将 GNN 迁移到超图结构上, 并提出了一系列超图神经网络方法. 我们按照是将超图展开成成对图结构还是直接在超图上进行卷积操作, 将这些方法分成展开型和非展开型两大类.

7.1 展开型超图GNN

图神经网络是基于成对图结构提出的, 并且已经衍生出一系列成熟的模型. 为了将图神经网络迁移到超图上, 一个直接的思路是先将超图按照某种固定模式转换成成对图, 再使用传统的 GNN 方法对其进行卷积. 根据展开方式的不同, 这些方法又可以进一步分为星形展开(star expansion)、团式展开(clique expansion)和线形展开(line expansion)3 种类型.

7.1.1 星形展开

如图 9(b)所示, 星形展开方式是指引入一个额外节点 i_e 代表超边 e , 并将 e 内所有节点与 i_e 相连, 从而将超图转换成一个新的成对图 \mathcal{G}^* . 对转换后的成对图中的每条边分配权重 $\omega^*(u, e) = \omega(e) / |e|$, 其中, $|e|$ 表示超边 e 中节点数量, 则 \mathcal{G}^* 的标准化拉普拉斯矩阵 \mathbf{L}^* 可表示为

$$\mathbf{L}^* = \begin{bmatrix} \mathbf{I} & -\tilde{\mathbf{A}} \\ -\tilde{\mathbf{A}}^T & \mathbf{I} \end{bmatrix} \quad (55)$$

其中, $\tilde{\mathbf{A}}$ 是 $|\mathcal{V}| \times |E|$ 矩阵:

$$\tilde{\mathbf{A}} = \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-\frac{1}{2}} \quad (56)$$

其中, $\tilde{\mathbf{A}}$ 中 u 行 e 列的值 $\tilde{\mathbf{A}}(u, e)$ 为

$$\tilde{\mathbf{A}}(u, e) = \frac{h(u, e)\omega^*(u, e)}{\sqrt{d^*(u)}\sqrt{d^*(e)}} \quad (57)$$

其中, $d^*(u)$ 和 $d^*(e)$ 分别表示超边内节点 u 和新代表超边的节点 e 的度. 具体地,

$$d^*(u) = \sum_{e \in \mathcal{E}} h(u, e)\omega^*(u, e), \quad d^*(e) = \sum_{u \in \mathcal{V}} h(u, e)\omega^*(u, e).$$

尽管星形展开和团式展开生成的成对图在节点的数量和连接上完全不同, Agarwal^[108]等人通过数学推导证明了 k 阶均匀超图的星形展开和团式展开的标准化拉普拉斯矩阵拥有相同的特征向量. 即使超图不是均匀的, 对星形展开图中每条边的权重扩大 $(|e|-1)|e|$ 后与团式展开仍具有相似的特征分解性质.

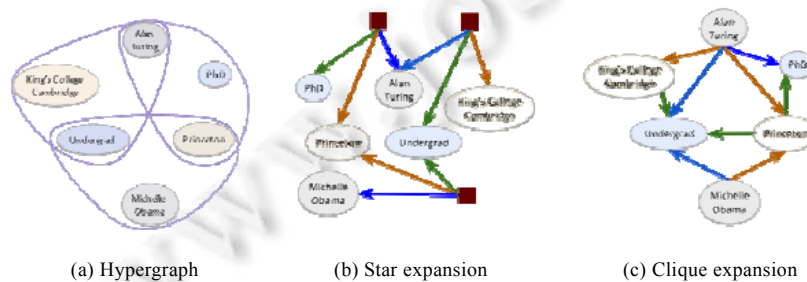


图 9 星形展开和团式展开^[104]

7.1.2 团式展开

(1) 超图拉普拉斯矩阵

如图 9(c)所示, 团式展开是指将超图中每一条超边转换成其内部节点两两相连的团结构(clique). 超图的团式展开策略最早追溯到 Zhou 等人^[100]关于超图最小切割问题的研究, 即对超图进行 k 次切割, 将其划分成 $k+1$ 个簇, 并使得所切割的边的权重和最小. 为此, Zhou 等人^[100]将对图上归一化割(normalized cut, N-cut)的概念^[109]扩展到超图上, 并提出了超图归一化割. 具体地, 一条超图割将超图 $\mathcal{G}^p = \{\mathcal{V}^p, \mathcal{E}^p\}$ 分成 S 和 S^c 两部分, 对于一条超边 $e \in \mathcal{E}^p$, 如果 e 中既存在节点属于 S , 又存在节点属于 S^c , 则称 e 被切割. 为了实现超图最小切割, 即要使得切割后产生的簇内部尽量紧密, 而簇之间被切割的超边的权值和尽量小, 因此超图归一化割被定义为

$$\mathcal{L}_{cut}(S) = vol \delta S \left(\frac{1}{vol S} + \frac{1}{vol S^c} \right) \tag{58}$$

其中, $vol S$ 表示簇中节点权重和 $vol S = \sum_{v \in S} d(v)$, $\delta S = \{e \in \mathcal{E}^p | e \cap S \neq \emptyset, e \cap S^c \neq \emptyset\}$ 表示被切割的超边集合, $vol \delta S$ 表示被切割的超边的权重和:

$$vol \delta S = \sum_{e \in \delta S} \omega(e) \frac{|e \cap S| |e \cap S^c|}{|e|} \tag{59}$$

公式(58)的最优化是一个 NP-complete 问题, 为了使其可求解, Zhou 等人^[100]将其放宽到实值优化问题, 并使用矩阵表达形式定义了超图拉普拉斯矩阵 L^p :

$$L^p = I - D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} \tag{60}$$

其中, D_v 和 D_e 分别表示节点和超边度矩阵, W 是超边权重矩阵. 类似于成对图, 公式(58)的求解转化成了求 L^p 最小非零特征值对应的特征向量. 公式(60)显式地推导出了超图拉普拉斯矩阵的形式, 由于其普适性, 后续从谱方法出发的超图卷积神经网络均对其进行了参考.

我们现在再次关注公式(59), 如图 10 所示, 如果使用团式展开的方式将两条待分割的超边转换成对应的全连子团, 并给子团中的每条普通边分配相同的权重 $\omega(e)/|e|$, $|e \cap S| |e \cap S^c|$ 则表示被切割的普通边数量. 因此, 公式(59)隐式地使用了团式展开将超图转化成了成对图. 由于这个原因, 我们将所有公式(60)衍生出的超图神经网络都归纳为团式展开方法.

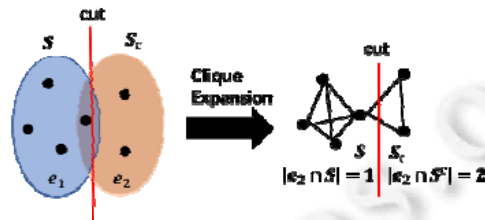


图 10 团式展开以及对应的超图割

(2) 团式展开方法

Feng 等人提出的 HGNN^[110]仿照成对图上关于图卷积的定义, 定义了超图卷积过程 $g_\theta * \mathbf{x} = U g_\theta U^T \mathbf{x}$. 其中, U 是超图拉普拉斯矩阵的特征矩阵, 即 $L^p = U \Lambda U^T$. 由于计算傅里叶变换的高复杂度, HGNN 同样引入切比雪夫多项式对卷积核 g_θ 进行 K 阶拟合, 即 $g_\theta * \mathbf{x} = \sum_{k=0}^K \theta_k T_k(\hat{\Delta}) \mathbf{x}$, 其中, $\hat{\Delta} = 2/\lambda_{\max} \cdot L^p - I$. 与 GCN 相似, 通过规定 $K=2$ 并进行一些参数限制, HGNN^[110]得到了如下的超图卷积层:

$$X^{(l+1)} = \sigma \left(D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} \right) X^{(l)} \Theta^{(l)} \tag{61}$$

其中, $\Theta^{(l)}$ 表示第 l 层卷积核, $\mathbf{X}^{(l)}$ 是第 l 层节点特征表示矩阵. 当超图退化成成对图时, 即每条边只连接两个节点, 有 $\mathbf{D}_e = 2\mathbf{I}$. 将其带入公式(61), 可以推导出:

$$\mathbf{X}^{(l+1)} = \sigma \left(\mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{W} \mathbf{H}^T \mathbf{D}_v^{-\frac{1}{2}} \right) \mathbf{X}^{(l)} \Theta^{(l)} = \sigma \left(\mathbf{D}_v^{-\frac{1}{2}} (\mathbf{D}_v + \mathbf{A}) \mathbf{D}_v^{-\frac{1}{2}} \right) \mathbf{X}^{(l)} \Theta^{(l)} = \sigma \left(\mathbf{I} + \mathbf{D}_v^{-\frac{1}{2}} \mathbf{A} \mathbf{D}_v^{-\frac{1}{2}} \right) \mathbf{X}^{(l)} \Theta^{(l)} \quad (62)$$

对比公式(62)和公式(16), 可以发现二者完全一致. 因此, 图神经网络可以视为超图神经网络的一种特殊形式.

HGNN^[110]首次成功地将图神经网络迁移到超图上并提出了一个普适性框架, 后续许多工作在此基础上进行了改进. Jiang 等人提出的 DHGNN^[111]考虑到超图的初始结构可能并不完善, 因此在每次卷积结束后会利用更新后的节点特征对超图进行重构. DHGNN^[111]由超图构造和超图卷积两部分组成. 其中, 超图构造过程利用 KNN 计算节点的 K 阶最近邻居从而构造超边. 在得到超图后将其输入到公式(61)所示的卷积层中进行学习, 输出的节点特征会再用于构造新的超图结构, 从而重复上述过程. Chen 等人^[112]将公式(16)与公式(61)相结合, 提出了一个同时处理成对图和超图的多级图神经网络. HGC-RNN^[113]则在 HGNN 的基础上添加 RNN 用于学习时序超图的特征表示, 与递归图神经网络(见第 4.1 节)做法相似, 它先是通过 HGNN 学习各个时间戳上超图的节点特征, 再使用 RNN 对各时刻特征进行编码, 并进行时序预测任务.

Bai 等人^[20]在 HGNN 的基础上进一步引入了注意力机制用以衡量不同节点在聚合过程中的权重. 与 GAT^[54]不同, Bai 等人提出的注意力网络并不是归纳式的, 而是在使用注意力机制时重新计算了关联矩阵 \mathbf{H} , 具体地:

$$\mathbf{H}_{ij} = \frac{\exp(\sigma(\text{sim}(\Theta \mathbf{x}_i, \Theta \mathbf{x}_j)))}{\sum_{k \in \mathcal{N}_i} \exp(\text{sim}(\Theta \mathbf{x}_i, \Theta \mathbf{x}_k))} \quad (63)$$

其中, \mathbf{x}_i 表示节点特征, Θ 表示卷积核, $\text{sim}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{a}^T [\mathbf{x}_i, \|\mathbf{x}_j]$ 表示节点特征相似度的计算. 这样, 关联矩阵 \mathbf{H} 中每一项的值不再是 0/1, 而是介于 0 与 1 之间的权重, 从而反映了不同节点和超边的重要性. 由于引入了注意力机制, \mathbf{H} 也变得可训练, 因此在每次反向传播中会得到更新. 图 11 描述了 Bai 等人模型的过程.

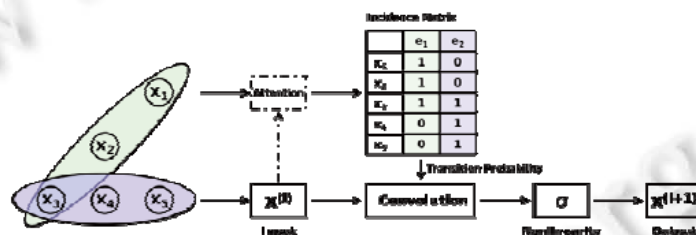


图 11 Bai 等人^[20]引入注意力机制后的超图卷积过程

(3) 团式展开的改进

团式展开方式由于使用一个稠密子图表示一条超边, 会产生很多新的成对边, 这在超边中节点数量较多时会带来较大复杂度. Yadati 等人^[21]指出了这个缺陷, 并提出了将团式展开简化的 HyperGCN^[21]. 根据对超边的不同展开方式, HyperGCN 具有多个版本. 如图 12 所示, HyperGCN 的一种展开方式是只使用超边中的某一对节点来表示这条超边, 即 1-HyperGCN^[21]. 它先对目标函数 $\arg \max_{i, j \in e} \|(\Theta^{(t)})^T (\mathbf{h}_i^{(t)} - \mathbf{h}_j^{(t)})\|_2$ 进行了最优化, 从而得到了每条超边中距离最远的两个节点 i_e 和 j_e , 并用节点 i_e 和 j_e 及其相连的边表示超边 e , 这样就可以使用 GCN 对转化后的成对图进行卷积: $\mathbf{h}_v^{(t+1)} = \sigma \left(\Theta^{(t)T} \sum_{\mu \in \mathcal{N}^{(v)}} (\mathbf{A}_{v, \mu}^{(t)} \cdot \mathbf{h}_\mu^{(t)}) \right)$.

但是很显然, 这种做法会造成许多节点和关系信息的损失, 从而使得模型收敛过慢且准确率较低. 因此, 完整的 HyperGCN 将超边中除了 i_e 和 j_e 的其他节点作为 mediator^[114], 并将 mediator 与 i_e 和 j_e 构成了一个二分图, 并用这个图表示原超边.

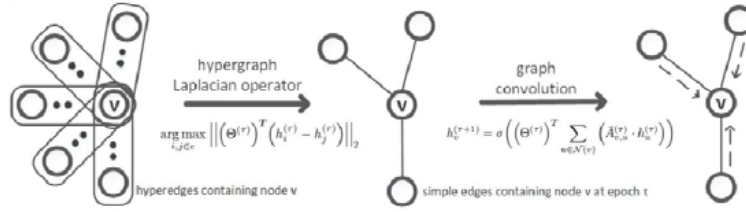


图 12 1-HyperGCN^[21]对超边进行线性展开并在此之上进行卷积

上述方法都是基于同质超图, Sun 等人^[19]关注了超图的异质性, 他们使用元路径将异质的成对图分成了一系列快照, 再基于这些快照构造异质超边. 这样, 原始的异质成对图被分解成多个包含不同语义信息的异质超图. 他们提出的 HWNN^[19]使用小波变换替代了傅里叶变换, 使得模型能够更好地学习局部信息. 相应地, 图卷积过程变成:

$$g_s * \mathbf{x}^{G_c} = \psi_s^{G_c} \Lambda^{G_c} (\psi_s^{G_c})^{-1} \mathbf{x}^{G_c} \quad (64)$$

其中, G_c 表示类型为 c 的异质超图, $\psi_s^{G_c}$ 表示小波变换, \mathbf{x}^{G_c} 是节点特征向量. 通过引入多项式级数对 Λ^{G_c} 进行拟合, 得到基于小波变换的图卷积层:

$$(\mathbf{X}^{G_c})^{(l+1)} = \sigma \left(\Theta_{\Sigma}^{G_c} \Lambda^{(l)} (\Theta_{\Sigma}^{G_c})' (\mathbf{X}^{G_c})^{(l)} \mathbf{W} \right) \quad (65)$$

其中, $\Theta_{\Sigma}^{G_c} = \sum_{k=0}^K \theta_k (\Theta^{G_c})^k$, $(\Theta_{\Sigma}^{G_c})' = \sum_{k=0}^{K'} \theta'_k (\Theta^{G_c})^k$, $\Theta = \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_v^{-\frac{1}{2}}$, $\Lambda^{(l)}$ 是对角过滤矩阵, 其参数可学习, \mathbf{W} 是节点特征投影矩阵, 将异质节点特征投影到同一空间. 在得到所有异质图上节点特征后, 通过连接操作得到最终的节点嵌入表示. 图 13 展示了 HWNN 的整体框架.

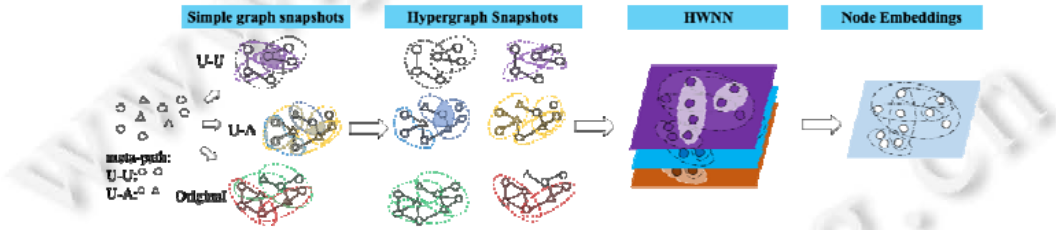


图 13 HWNN^[19]异质超图构造及卷积过程

在超边预测任务方面, NHP^[115]采用了构造对偶超图^[116]的方法将超边预测任务转换成对偶超图的节点分类任务. 它使用团式展开将构造的对偶超图转换成成对图, 再使用 GCN^[9]聚合节点特征并进行节点分类.

7.1.3 线形展开

Yang 等人^[117]指出星形展开和团式展开均存在一定缺陷. 例如, 星形展开后, 原本同质的超图变成了异质图, 而经典图神经网络多是对同质图的; 团式展开则会造成超边与节点间的关系丢失. 鉴于这两种展开方式所存在的缺陷, Yang 等人^[117]提出了线形展开(line expansion)的方式将超图 G^p 转换成成对图 G' . 线形展开的基本方式是将每个节点与包含它的超边组成一个新的节点, 称为线点. 如图 14 所示, (v_3, e_2) 组合构成了一个线点. 对任意两个线点 $S=(v, e)$ 和 $S'=(v', e')$, 如果 $v=v'$ 或 $e=e'$, 则 S 与 S' 间存在一条边. 很显然, G' 具有两种类型的边, Yang 等人^[117]因此设计了基于节点相似性和基于超边相似性的两种信息传递方式, 对应的卷积层如下:

$$\mathbf{h}_{(v,e)}^{(l+1)} = \sigma \left(\sum_e \mathbf{W}_e \mathbf{h}_{(v,e)}^{(l)} \Theta^{(l)} + \sum_{v'} \mathbf{W}_v \mathbf{h}_{(v',e)}^{(l)} \Theta^{(l)} \right) \quad (66)$$

其中, \mathbf{W}_e 和 \mathbf{W}_v 分别表示超边相似性和节点相似性的系数矩阵, $\mathbf{h}_{(v,e)}^{(l)}$ 是线点特征向量, $\mathbf{h}_{(v,e)}^{(0)} = \mathbf{x}_v$.

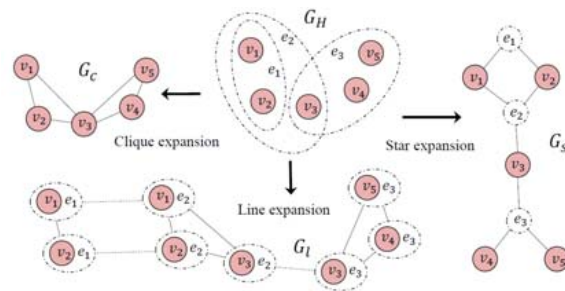


图 14 线形展开过程^[117]

7.2 非展开型超图GNN

展开型方法虽然基于严谨的数学推导, 但存在两大缺陷: (1) 无论是哪种展开方式, 将超图转换成成对图的过程都会损失一部分原有的语义信息; (2) 超图的构建是为了表示成对图难以表征的高阶关联, 而展开型方法又将其还原为成对图, 这样的做法显然并不合理. 因此, 一些研究者不再将超图展开成成对图, 而是直接在超图上进行卷积操作.

Arya 等人提出的 HyperSage^[22]不再依靠公式(61)定义的谱方法对超图进行卷积, 而是参考了 GraphSage^[11]等空间方法沿着图结构进行信息传递的思想. 与成对图中节点信息直接通过相连边进行传递不同, HyperSage 采用了一个两级聚合过程, 即先聚合同一超边中的节点特征生成超边特征, 再聚合包含中心节点的超边特征, 得到中心节点的嵌入表示. 与 GraphSage^[11]相似, HyperSage 使用了采样机制获取固定数量的邻居节点. 具体地, $\mathcal{N}(v_i, e, \alpha)$ 表示超边 e 内的节点采样, 其中, α 表示采样数量; $E(v_i)$ 表示包含中心节点 v_i 的超边集合. HyperSage 的两级聚合过程如下:

$$\mathbf{x}_{j,l}^{(e)} \leftarrow \mathcal{F}_1(\{\mathbf{x}_{k,l-1} | v_k \in \mathcal{N}(v_i, e, \alpha)\}) \quad (67)$$

$$\mathbf{x}_{i,l} \leftarrow \mathbf{x}_{i,l-1} + \mathcal{F}_2(\{\mathbf{x}_{j,l}^{(e)} | e_j \in E(v_i)\}) \quad (68)$$

其中, $\mathbf{x}_{j,l}^{(e)}$ 表示将 $\mathcal{N}(v_i, e, \alpha)$ 内节点特征聚合后得到的超边特征, $\mathcal{F}_1(\cdot)$ 和 $\mathcal{F}_2(\cdot)$ 分别表示节点聚合函数和超边聚合函数, 其具体形式为

$$\mathcal{F}_1(v_i) = \left(\frac{1}{|\mathcal{N}(v_i, e)|} \sum_{v_k \in \mathcal{N}(v_i, e)} \mathbf{x}_k^p \right)^{\frac{1}{p}}, \quad \mathcal{F}_2(v_i) = \left(\frac{1}{|E(v_i)|} \sum_{e \in E(v_i)} \frac{\mathcal{N}(v_i, e)}{\mathcal{N}(v_i)} (\mathcal{F}_1(v_i))^p \right)^{\frac{1}{p}} \quad (69)$$

其中, 幂次 p 作为超参数可有多种取值, 如 $p=1, p=0.01$ 或 $p=-1$. HyperSage 的特征聚合过程如图 15 所示.

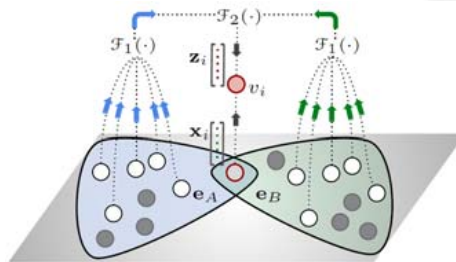


图 15 HyperSage^[22]的两级聚合过程

能够直接在超图结构上进行卷积, 由于借鉴了 GraphSage 的思想, HyperSage 同时也继承了 GraphSage 的诸多优点. 作为归纳式方法, HyperSage 不需要像 HGNN 那样一次性输入超图结构, 在图结构发生变化时也只需使用变化部分重新训练模型. 同时小批量训练也成为可能, 使得模型可应用于更大规模的超图网络. 关于归纳式模型优势的详细阐述可回顾第 4.3.2 节.

Ding 等人提出的 HyperGAT^[118]进一步引入了注意力机制衡量不同特征在聚合过程中的重要性. 与 Bai 等

人^[20]直接使用注意力系数更新关联矩阵的方式不同, HyperGAT 采用了一个两级注意力机制分别衡量节点和超边特征:

$$\mathbf{f}_j^{(l)} = \sigma \left(\sum_{v_k \in e_j} \alpha_{jk} \mathbf{W}_1 \mathbf{h}_k^{(l-1)} \right), \mathbf{h}_i^{(l)} = \sigma \left(\sum_{e_j \in \mathcal{E}_i} \beta_{ij} \mathbf{W}_2 \mathbf{f}_j^{(l)} \right) \quad (70)$$

其中, α_{jk} 和 β_{ij} 分别表示节点和超边的注意力系数.

上述方法主要针对节点分类任务, 在超边预测方面也提出了诸多非展开式方法. 其中, Tu 等人^[23]指出一些超边具有不可分解性, 他们提出的 DHNE^[23]使用自编码器对 $\mathbf{H}\mathbf{H}^T - \mathbf{D}_v$ 进行了重建, 并使用重建节点特征计算节点间相似性, 从而预测超边存在的概率. 虽然 DHNE 不需要对超边进行展开, 但只适用于超边大小固定的匀质超图, 因此有较大的应用限制.

Zhang 等人提出的 Hyper-SAGNN^[119]则适用于任意大小的异质超边. 对于一组节点 $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, Hyper-SAGNN 分别使用多头注意力和线性映射方式得到了节点的动态表示 $\{\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n\}$ 和静态表示 $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n\}$. 通过哈达玛积计算两组表示之间的距离可以得到这组节点之间的相似性, 进而预测它们构成一条超边的概率.

非展开型方法由于不需要将超图重新还原成成对图, 从而避免了信息损失, 保留了节点间的完整语义关系. 同时, 一些非展开方法可以通过小批量训练, 使得模型可以应用于大规模超图网络. 这些原因使得非展开方法成为当前超图学习的研究热点之一.

完整的超图神经网络方法的分类和描述见表 4.

表 4 超图神经网络总结

方法	类型	算法描述	缺陷	应用
HGNN ^[110]	团式展开	使用 L^p 推导谱方法形式的超图卷积网络	当超图中存在孤立点时 L^p 会失效	节点分类 视觉目标分类
Bai, <i>et al.</i> ^[20]		引入 attention 机制进行关联矩阵更新	不适用于节点和超边异质的情况	
DHGNN ^[111]		每层卷积之后对超图进行重构	超图的更新导致复杂度较高	
Chen, <i>et al.</i> ^[112]		将成对图与超图相结合并进行层次卷积	模型复杂度较高不适用于大规模网络	
HGC-RNN ^[113]		使用 HGNN 学习各时刻节点特征再通过 RNN 编码	模型可训练参数过多	
HyperGCN ^[21]		使用最大距离节点对简化团式展开	简化展开过程会造成信息损失	
HWNN ^[119] NHP ^[115]		使用小波变换替换傅里叶变换 构建对偶超图并使用超图 GNN 进行卷积	模型复杂度较高 不是归纳式模型	
Yang, <i>et al.</i> ^[117]	线形展开	将节点和超边对定义为线点, 并通过节点和超边相似性分别定义线点的边	模型可迁移性较差	节点分类 视觉目标分类
HyperSage ^[22]	非展开	使用采样机制进行两级特征聚合	抽样过程造成部分局部信息丢失	节点分类
HyperGAT ^[118]		引入 attention 机制在两级聚合过程分配注意力权重	计算注意力时未考虑超边本身的属性特征	文本分类
DHNE ^[23] Hyper-SAGNN ^[119]		使用自编码器对超图拉普拉斯矩阵重建 引入 attention 机制计算超边存在概率	只适用于匀质超图 随着超边数量的增长, 模型复杂度迅速增加	超边预测

8 应用

复杂图结构相比于简单图减少了诸如同质、静态等限制条件, 使得它们更加贴近实际情况, 进而拥有更加广泛的应用场景. 基于复杂图的 GNN 在不同领域都有着广泛应用, 例如时空图神经网络在交通流预测^[95,96]和动作识别^[18]领域; 异质图神经网络在推荐系统^[57,58,71]、视觉问答^[65]和金融风控^[120]领域; 或者超图神经网络在视觉^[121,122]和化学领域^[115]的重要作用. 除此之外, 一些实际应用则对多种复杂图进行了组合使用, 例如时

序知识图谱^[93]和超图知识图谱^[104]的相关研究. 本节我们按照不同类型复杂图及其组合方式阐述复杂图 GNN 的相关应用方向.

8.1 异质图

8.1.1 推荐系统

异质图神经网络被广泛用于推荐系统的辅助工作. 为了表示用户和物品间的关系, 研究者往往会构造用户-物品的关联图, 并进一步融合知识图谱以增加关联图的深度和广度. 相比于知识图谱补全或事实推理等应用, 推荐系统的图谱规模相对较小, 因此一些研究者开始将 GNN 应用到 KG 的表示学习上. 其中, Wang 等人^[71]提出一个端到端的知识图谱卷积网络 KGCN, 并将学习到的节点表示辅助推荐系统. KGAT^[58]将知识图谱与用户-物品二分图相融合, 并使用多层图注意力网络^[54]对融合后的结果进行学习. Huang 等人^[123]和 Ma 等人^[124]使用基于元路径的异质图神经网络进行多种内容推荐.

8.1.2 金融风控

在金融领域, 近年来有不少研究者开始尝试将异质图神经网络应用于欺诈检测、团伙识别等金融风控应用, 并取得了令人瞩目的成果. Liu 等人^[125]构造了一个包含账户和登录设备两种类型节点的异质图, 并通过系统标注一部分有欺诈风险的节点. 他们提出的 GeniePath^[125]通过结合使用自适应深度优先和广度优先搜索, 只对潜在风险传播路径上的节点特征进行聚合, 进而识别出潜在风险节点. Liu 等人提出的 GEM^[120]在账户-设备异质图上进一步引入历史行为数据作为账户的状态向量, 并设计了一个异质图注意力网络辨识风险节点和正常节点.

尽管图神经网络开始在互联网金融领域进行应用, 但 GNN 的不可解释性和较高的复杂度仍旧限制了它在信贷风险辨识和股票风险预测等传统金融风控领域发挥作用. 为了提高模型的可解释性, 一些研究也开始结合 Bayes 网络等概率图模型来解决金融风控问题^[126,127].

8.1.3 计算机视觉

异质图神经网络在计算机视觉中的应用包括视觉问答^[128,129]和语义分割^[130]等. 其中, 视觉问答(visual question answering, VQA)是指通过图片信息以及关于图片的问题文本信息给出相应的答案的任务. 视觉问答需要融合图片中不同目标间的空间关系和目标本身的语义信息来回答问题, 因此以图片中的目标作为节点构造异质图是 VQA 中常见的做法. Teney 等人^[129]最早在 VQA 中引入图神经网络. 他们根据图像场景和问题句法分别构造了两个异质图, 并使用循环图神经网络学习节点特征并用于生成答案. Jiang 等人^[65]使用视频中的不同帧和问题中的不同词组作为节点构造异质图, 并将帧节点和词节点对齐, 通过 GCN 学习节点特征用于答案生成. 最近, Gao 等人^[131]抽取了图像中数字、文本和视觉信息, 将它们分别作为节点构造了 3 个网络. 通过图注意力网络对 3 张图间的信息进行交叉聚合, 获取了更加复杂的语义交互信息用于视觉问答.

在语义分割方面, 该任务要求为图像中每个像素分配一类标签, 从而将整张图像分割成具有不同语义信息的几个区域. 在分类过程中往往需要考虑全局语义信息, 因此一些研究将图像中不同区域间的关系抽象成图结构. Liang 等人^[130]最早根据区域间的距离关系构造图结构, 并使用递归图神经网络聚合邻居信息. Te 等人^[132]和 Landrieu 等人^[133]为了解决点云分割问题, 通过点云坐标值和颜色等特征值计算相似性, 将相似点抽象成超点进而构造超点网络, 并使用 GCN 对超点网络实施卷积.

8.2 动态图

8.2.1 链接预测

动态图的一个典型应用是链接预测任务. 通过动态图结构的历史变化情况, 预测将来一段时间内链接存在的概率. 基于 GNN 的动态图链接预测问题大多使用 GNN 学习某个时间戳上的图特征再输入到 RNN 中对所有时间点编码. 例如, Chen 等人^[134]最早将 GNN 与 LSTM 结合用于动态图链接预测. Lei 等人^[135]和 Yang 等人^[136]将 GCN 与 GAN 相结合对动态图中空间和时间特征进行学习. Pareja 等人^[35]进一步考虑了动态图节点集在不同时间戳发生变化的情况. 图的动态性广泛存在于各种领域, 因此动态图链接预测在社交网络^[137]、知识

图谱^[17]以及生物网络^[138]中都具有重要意义。

8.2.2 交通流

交通流问题中通常将城市道路抽象成网络结构,其中节点表示路口上的传感器,边表示传感器间的相互距离。传感器可以探测到一段时间内的交通速度,并将其作为节点的动态特征。交通流预测问题便是在给定网络的空间结构和一段时间内的交通速度下,预测下一时刻的交通速度。Zhang 等人^[83]最早使用门控注意力网络学习交通网络中的时间和空间信息。Yu 等人^[95]进一步提出时空图卷积网络 STGCN,通过一个统一框架同时对时间和空间信息进行卷积,相比于将 GNN 作为循环网络单元的做法,有效降低了模型复杂度。Guo 等人^[96]则在此基础上进一步引入了注意力机制。最近,时空图神经网络也被用于交通事故预测^[139]、共享单车需求预测^[140]以及融合导航数据的交通预测^[141]等其他交通相关应用。

8.2.3 动作识别

时空图神经网络还被应用于人体骨骼的动作识别任务。给定一段人物视频,通过将人体骨骼抽象成网络结构,在不同时间戳下,随着人体的运动,网络空间结构会发生相应变化。Jian 等人提出的 Structural-RNN^[15]最早将 RNN 与 GNN 相结合对动作模式进行分类。Yan 等人^[18]则最早使用时空图神经网络进行人体动作识别。使用时空图神经网络对骨骼动作识别已经成为近几年的研究热点,并衍生出一系列方法^[94,97,142]。

8.3 超图

8.3.1 化学

超图在化学领域有着广泛应用,其中无向超图可以建模化学分子结构,而有向超图则可以表示化学反应过程。Konstantinova 等人^[143]指出,拥有多中心离域键的非典型结构的化合物分子不能用成对图结构很好地加以表示。他们进而引入超图作为这些化合物的数学模型并给出了相应的形式化定义。Yadeti 等人^[115]则使用有向超图表示化学反应过程。有向超图由一系列超边构成,每对超边由一条有向边定义了方向。因此,化学反应可以用一条有向超边表示,其中,反应物和生成物对应超边对,超边方向表示反应方向。Yadeti 等人^[115]进一步使用超图 GNN 对有向超图进行卷积,将得到的节点特征用于化学反应超边的链接预测。

8.3.2 计算机视觉

超图 GNN 在计算机视觉领域的应用包括三维物体分类^[122,144]和多模态数据处理^[121,145]等。在三维物体分类方面,研究者使用多张超图表示三维物体的不同二维视图。在超图中,节点表示不同的物体,超边表示视角聚类后的一个簇。这样,融合多张超图的结构信息即可重构出三维物体的特征,用于物体辨识。Gao 等人^[122]最早使用团式展开方法对多个超图进行卷积并用于三维物体辨识。Zhang 等人^[144]进一步使用一个归纳式超图卷积网络以降低直推式方法在测试新数据上的复杂度。除了三维物体外,超图神经网络还被用于三维点云的分析^[146]。

在多模态数据方面,研究者通过构建不同类型的超图对多模态数据进行更好的融合。例如, Kim 等人^[145]使用超图注意力网络学习语义超图以解决多模态数据间信息水平不对称的问题。Chen 等人^[121]将推文作为节点,使用超边建模推文在不同模态下的相似度,通过对超图进行直推式学习进行微博情感预测。Zhang 等人^[147]使用超图表示多模态数据间高阶关联,通过多任务学习进行阿尔茨海默症的多模态分类。

8.4 复杂图的组合应用

在另外一些应用中同时涉及到多种复杂图结构。例如,动态知识图谱学习同时涉及动态图和异质图;超图知识图谱以及一些社交网络应用同时涉及异质图与超图。我们按照复杂图的组合方式分别介绍这些应用。

8.4.1 动态异质图

这类应用主要集中在动态知识图谱上。由于知识库中事实关系会随时间发生变化,反映在知识图谱上即实体或关系的增删或变更。如果将每次图谱更新对应一个时间戳,就相应地得到了动态知识图谱。动态知识图谱的生成以及学习在推荐系统、问答系统以及金融风控等具有实时要求的应用中具有重要意义。早期相关研究主要集中在动态知识图谱的生成上。Carlson 等人^[148]最早使用一个信息提取器从文本中提取新的事实三

元组, 并添加到知识图谱中, 每个事实会被分配一个置信概率值. Dong 等人^[149]在此基础上使用图谱中已经存在的事实关系计算新的事实先验概率. 近年来, 研究者开始关注动态知识图谱的表示学习问题. 知识图谱往往具有巨大的规模, 而在每个时间戳变化的实体和关系往往只占知识图谱很小的一部分. 因此在不影响特征表示质量的前提下如何尽量降低重新训练图谱的复杂度是该问题的关键. Wu 等人^[93]提出了一个基于 GCN 的动态知识图谱表示学习模型, 通过将知识图谱更新将表示向量的影响限制在一定区域内, 该模型支持对图谱表示的在线学习. 另外, Tang 等人^[150]提出一个时间跨度感知的动态知识图谱表示学习模型.

8.4.2 异质超图

异质超图被广泛应用于社交网络挖掘. 社交网络中往往包含用户、贴文和评论等多种类型的节点以及相应的多种关系, 是一种典型的异质结构. 大多数社交网络研究都使用成对关系描述用户与用户或用户与评论间的直接关联. 然而, 近来 Chen 等人^[112]指出社交网络中许多高阶关联更适合使用超图进行表示. 超边的灵活定义方式使其可以表示各种语义关系. 例如, 使用用户 N 阶邻居定义的超边可以反映该用户的朋友圈, 使用中心性定义的超边可以反映用户的社交重要性. Tan 等人^[101]较早地在社交网络的用户对齐任务中引入了超图, 用以建模用户间的高阶关联. 在此基础上, Chen 等人^[112]将成对图与超图相结合, 利用二者在描述用户关系上的互补优势, 提出了一个多级卷积神经网络对跨网络用户进行对齐.

除此之外, Fatemi 等人^[151]将知识图谱从三元组结构扩展到了超图结构, 提出了知识超图的概念. 他们还将 Simple^[70]从三元组知识图谱扩展到了知识超图, 提出了一个知识超图表示学习模型^[151].

9 数据集

复杂图 GNN 由于其广泛的应用场景拥有众多公开数据集. 我们按照复杂图种类以及具体的应用类型分别介绍相应的数据集. 由于数据集数量众多, 本文只介绍各应用中被广泛使用的一部分数据集. 对于想获取更多其他复杂图 GNN 相关数据集的读者, 可以访问 Network Repository (<http://networkrepository.com>)、Stanford Large Network (<https://snap.stanford.edu/data/>)、Open Graph Benchmark (<https://ogb.stanford.edu>)以及 LINQS (<https://linqs.soe.ucsc.edu/data>)等图数据仓库. 本文介绍的数据集分类及相应描述见表 5.

表 5 复杂图神经网络数据集总结

网络类型	应用	分类	数据集
异质图	节点分类, 链接预测, 节点聚类	引文网络 电影网络 社交网络 生物网络	DBLP ^[152] , AMiner ^[154] , ACM ^[153] IMDB ^[157] , MovieLens ^[158] Blogcatalog ^[159] , Yelp ^[160] PubMed ^[161]
	推荐系统, 视觉问答	知识图谱 用户商品网络	Freebase ^[162] , Wikidata ^[165] , YAGO ^[166] Last.fm ^[175] , Amazon-book ^[176]
动态图	链接预测	引文网络 社交网络	HEP-TH Social Evolution ^[167] , Github ^[168] , Reddit ^[177] , UCI ^[169]
	交通流预测	交通网络	PeMSD4 ^[170] , PeMSD8 ^[170] , BJR4 ^[96]
	动作识别	骨骼网络	Kinetics ^[171] , NTU-RGB+D ^[18]
超图	节点分类, 链接预测, 节点聚类	引文网络 蛋白质网络	Cora ^[21] , Citeseer ^[21] , Pumbed ^[21] Yeast ^[172]
	视觉分类	3D 物体网络	NTU ^[173] , ModelNet40 ^[174]

在异质图方面, 普通异质图的应用集中在节点分类、链接预测以及节点聚类等方向上. 其中, DBLP^[152]、ACM^[153]和 AMiner^[154]等引文网络被广泛用于这些应用的验证. 不同于 Cora^[155]等同质引文网络, 这些引文网络都具有文献、作者以及会议等多种类型节点, 节点特征通常由论文文本通过 word2vec^[156]转换的词向量构成. 除此之外, IMDB^[157]以及 MovieLens^[158]等电影网络具有电影、演员以及导演等多种类型节点, 节点特征由电影情节转换的词向量构成. 在社交网络方面, Blogcatalog^[159]是一个博客网络, 包含用户和组群两种节点, 和用户-用户以及用户-组群两种类型的边; Yelp^[160]是一个社交评价网络, 包含用户对美国各大城市商铺的评

价,具有用户、商铺、城市以及商业类型等几种节点.生物信息方面,PubMed^[161]是基因疾病网络,包含基因、疾病、化合物以及物种等多种类型节点.

对于知识图谱,一方面,知识图谱数据集可以作为异质图进行节点分类和链接预测等任务;另一方面,可以作为辅助信息进行商品推荐或视觉问答等工作.常用的知识图谱数据集是几个大型的公开数据集.其中,Freebase^[162]是一个结构化知识库,拥有图书、电影、音乐以及人物等各种类型实体,按照实体数量具有FB13^[163]、FB15K^[164]等多个数据集.此外,Wikidata^[165]和YAGO^[166]等开源知识图谱都是基于维基百科构造的.

动态图同样具有众多公开数据集.时序图方面,HEP-TH是一个动态引文网络,搜集了arXiv网站从1993年1月到2003年4月部分论文间相互引用的情况.社交网络方面,Social Evolution^[167]包含了83个用户从2008年1月到9月间社交关系的变更;Github^[168]包含了2013年1月到2013年12月期间Github网站用户间关注、收藏等关系的变更.UCI^[169]是加州大学学生间动态社交关系网络,节点包含学生和社团,边表示学生与社团间关联.

时空图方面,PeMSD4^[170]和PeMSD8^[170]是美国加州高速公路交通流数据,采样时间间隔为30s.BJER4^[196]是北京市东四环主要区域交通流数据.Kinetics^[171]是人体动作数据集,包含400种人体动作.NTU-RGB+D^[18]是人体动作识别数据集,包含60种动作类型.

超图方面,大多数超图数据集都是通过人为来定义超边,从成对图转化过来的.例如,Cora^[21]、Citeseer^[21]和Pumbed^[21]等引文网络是通过定义共同引用关系或共同作者关系作为超边而得到的.Yeast^[172]酵母菌蛋白网络通过K近邻算法构造超图.计算机视觉方面,NTU^[173]和ModelNet40^[174]是多视角超图数据集,用于对三维物体进行分类.

10 展望与总结

10.1 研究方向展望

GNNs目前已经完成了向异质图、动态图和超图等复杂图结构的迁移,并取得了优秀的成果.但值得注意的是,当前大多数GNN方法只在小型数据集上进行了简单任务(节点分类、链接预测等)的验证.在真实应用场景下,复杂图的巨大规模和快速更新频率对复杂图GNN进一步提出了可扩展、在线性等要求.本节将讨论复杂图GNN几个潜在的研究方向.

- 可扩展性.当前用于复杂图GNN测试的数据集数据量都不大,大多在1万节点以下.然而和众多深度学习模型一样,GNNs在大规模图结构上的训练仍非常具有挑战性.一方面,随着GNN层数的增加,模型训练所需要的时间将呈指数增长;另一方面,大规模图的邻接矩阵以及节点特征储存需要消耗大量的内存空间.近年来,Cluster-GCN^[178]等研究考虑到了GNNs在大规模图上扩展性的问题,将训练的图规模提升到百万节点级别.但GNN在千万级节点以上的知识图谱,动态图上的训练仍是需要亟待解决的问题.

- 可解释性.随着GNN在金融风控、疾病诊断等领域的应用,对GNN可解释性的要求也越来越高.随着可解释性研究在视觉和文本领域的重大进展,近来一些研究者也开始从模型或梯度等角度给出神经网络的可解释性^[179].GNN模型的可解释性目前的研究还很不完善,是一个极具潜力的研究方向.

- 在线性.在动态网络方面,在欺诈检测或推荐系统实际应用中,用户行为网络的高频更新对GNN模型的响应时间提出了要求.如何在极短时间内重新训练大规模复杂图从而得到新的节点嵌入表示是一个重要的研究方向.近年来,一些研究者开始注意到这个问题.例如,Wu等人^[93]通过只训练变化的子图结构提出了一个大型知识图谱的在线学习模型.

- 迁移性.RNN被用于动态图GNN以及简单图GNN向超图GNN的迁移已经证明了深度学习模型在不同图结构上的可迁移性.一些原本针对某种图类型设计的方法经过简单修改和拓展后很可能用于其他图结构.复杂图GNN之间的迁移研究是将来重要的研究方向之一.

10.2 总 结

基于复杂图的图神经网络是近年来的研究热点. 本文将复杂图分为异质图、动态图和超图这 3 种基本类型. 针对每一种复杂图, 本文给出了全面、系统的分类框架, 分别比较了同类算法间的优点与缺陷, 并详细阐述了不同类算法间的思路差异. 本文还讨论了 3 种复杂图各自的以及相互组合后的应用领域, 并列举了一些具有代表性的复杂图公开数据集. 最后, 本文对复杂图神经网络未来可能的研究方向进行了展望.

References:

- [1] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations. In: Macskassy SA, ed. Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining (SIGKDD). New York: ACM, 2014. 701–710.
- [2] Grover A, Leskovec J. node2vec: Scalable feature learning for networks. In: Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining. New York: ACM, 2016. 855–864.
- [3] Sperduti A, Starita A. Supervised neural networks for the classification of structures. *IEEE Trans. on Neural Networks*, 1997, 8(3): 714–735.
- [4] Gori M, Monfardini G, Scarselli F. A new model for learning in graph domains. In: Jiang T, ed. Proc. of the IEEE Int'l Joint Conf. on Neural Networks. Piscataway: IEEE, 2005. 729–734.
- [5] Gallicchio C, Micheli A. Graph echo state networks. In: Jiang T, ed. Proc. of the IEEE Int'l Joint Conf. on Neural Networks (IJCNN). Piscataway: IEEE, 2010. 1–8.
- [6] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G. The graph neural network model. *IEEE Trans. on Neural Networks*, 2008, 20(1): 61–80.
- [7] Xu B, Shen H, Cao Q, Qiu Y, Cheng X. Graph wavelet neural network. In: Sainath T, ed. Proc. of the Int'l Conf. on Learning Representations (ICLR). New Orleans: OpenReview, 2019.
- [8] Defferrard M, Bresson X, Vandergheynst PJ. Convolutional neural networks on graphs with fast localized spectral filtering. In: Lee DD, ed. Proc. of the Conf. on Neural Information Processing Systems. Cambridge: MIT Press, 2016.
- [9] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: Sainath T, ed. Proc. of the Int'l Conf. on Learning Representations (ICLR). New York: OpenReview, 2017. 1–14.
- [10] Gilmer J, Schoenholz SS, Riley PF, Vinyals O, Dahl GE. Neural message passing for quantum chemistry. In: Jebara T, ed. Proc. of the Int'l Conf. on Machine Learning (ICML). New York: ACM, 2017. 1263–1272.
- [11] Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. In: Guyon I, ed. Proc. of the Conf. on Neural Information Processing Systems (NIPS). Cambridge: MIT Press, 2017. 1024–1034.
- [12] Schlichtkrull M, Kipf TN, Bloem P, Van Den Berg R, Titov I, Welling M. Modeling relational data with graph convolutional networks. In: Gangemi A, ed. Proc. of the European Semantic Web Conf. Springer, 2018. 593–607.
- [13] Hu Z, Dong Y, Wang K, Sun Y. Heterogeneous graph transformer. In: Huang Y, ed. Proc. of the Web Conf. (WWW). New York: ACM, 2020. 2704–2710.
- [14] Wang X, Ji H, Shi C, Wang B, Ye Y, Cui P, Yu PS. Heterogeneous graph attention network. In: Huang Y, ed. Proc. of the Web Conf. (WWW). New York: ACM, 2019. 2022–2032.
- [15] Jain A, Zamir AR, Savarese S, Saxena A. Structural-RNN: Deep learning on spatio-temporal graphs. In: Bajcsy R, ed. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2016. 5308–5317.
- [16] Mohanty S, Pozdnukhov A. Graph CNN+ LSTM framework for dynamic macroscopic traffic congestion prediction. In: Fakhraei S, ed. Proc. of the Int'l Workshop on Mining and Learning with Graphs. New York: ACM, 2018. 1–14.
- [17] García-Durán A, Dumančić S, Niepert M. Learning sequence encoders for temporal knowledge graph completion. In: Riloff E, ed. Proc. of the Conf. on Empirical Methods in Natural Language Processing (EMNLP). Stroudsburg: ACL, 2018.
- [18] Yan S, Xiong Y, Lin D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2018. 1–9.
- [19] Sun X, Yin H, Liu B, Chen H, Cao J, Shao Y, Viet Hung NQ. Heterogeneous hypergraph embedding for graph classification. In: Carmel D, ed. Proc. of the ACM Int'l Conf. on Web Search and Data Mining. New York: ACM, 2021. 725–733.
- [20] Bai S, Zhang F, Torr PH. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 2021, 110(1): 107637.

- [21] Yadati N, Nimishakavi M, Yadav P, Nitin V, Louis A, Talukdar P. HyperGCN: A new method of training graph convolutional networks on hypergraphs. In: Guyon I, ed. Proc. of the Conf. on Neural Information Processing Systems (NIPS). Cambridge: MIT Press, 2019.
- [22] Arya D, Gupta DK, Rudinac S, Worring M. HyperSAGE: Generalizing inductive representation learning on hypergraphs. CoRR, 2020.
- [23] Tu K, Cui P, Wang X, Wang F, Zhu W. Structural deep embedding for hyper-networks. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2018. 1–10.
- [24] Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY, Systems I. A comprehensive survey on graph neural networks. IEEE Trans. on Neural Networks, 2021, 32(1): 4–24.
- [25] Zhang Z, Cui P, Zhu W, Engineering D. Deep learning on graphs: A survey. IEEE Trans. on Knowledge, 2022, 34(1): 249–270.
- [26] Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun MJAO. Graph neural networks: A review of methods and applications. AI Open, 2020, 1: 57–81.
- [27] Jiang W, Luo JJ. Graph neural network for traffic forecasting: A survey. arXiv preprint arXiv: 2101.11174, 2021.
- [28] Wu S, Sun F, Zhang W, Cui BJ. Graph neural networks in recommender systems: A survey. arXiv preprint arXiv: 2011.02260, 2020.
- [29] Frasconi P, Gori M, Sperduti A. A general framework for adaptive processing of data structures. IEEE Trans. on Neural Networks, 1998, 9(5): 768–786.
- [30] Powell MJ. An efficient method for finding the minimum of a function of several variables without calculating derivatives. The Computer Journal, 1964, 7(2): 155–162.
- [31] Pineda FJ. Generalization of back-propagation to recurrent neural networks. Physical Review Letters, 1987, 59(19): 2229.
- [32] Li Y, Tarlow D, Brockschmidt M, Zemel RJ. Gated graph sequence neural networks. In: Sainath T, ed. Proc. of the Int'l Conf. on Learning Representations (ICLR). New York: Open Review, 2015. 1–20.
- [33] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Riloff E, ed. Proc. of the Empirical Methods Natural Language Processing (EMNLP). Stroudsburg: ACL, 2014. 1724–1734.
- [34] Dai H, Kozareva Z, Dai B, Smola A, Song L. Learning steady-states of iterative algorithms over graphs. In: Jebara T, ed. Proc. of the Int'l Conf. on Machine Learning (ICML). New York: ACM, 2018. 1106–1114.
- [35] Pareja A, Domeniconi G, Chen J, Ma T, Suzumura T, Kanezashi H, Kaler T, Schardl T, Leiserson C. Evolvegn: Evolving graph convolutional networks for dynamic graphs. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2020. 5363–5370.
- [36] Qiu J, Dong Y, Ma H, Li J, Wang K, Tang J. Network embedding as matrix factorization: Unifying deepwalk, line, PTE and node2vec. In: Carmel D, ed. Proc. of the ACM Int'l Conf. on Web Search and Data Mining. New York: ACM, 2018. 459–467.
- [37] Wang D, Cui P, Zhu W. Structural deep network embedding. In: Krishnapuram B, ed. Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining (KDD). New York: ACM, 2016. 1225–1234.
- [38] Cao S, Lu W, Xu Q. Deep neural networks for learning graph representations. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2016. 1–10.
- [39] Berg RVD, Kipf TN, Welling M. Graph convolutional matrix completion. arXiv preprint arXiv: 02263, 2017.
- [40] Kingma DP, Welling M. Auto-encoding variational bayes. In: Sainath T, ed. Proc. of the Int'l Conf. on Learning Representations (ICLR). New York: Open Review, 2014. 1–10.
- [41] Kipf TN, Welling M. Variational graph auto-encoders. In: Gal Y, ed. Proc. of the NIPS Workshop on Bayesian Deep Learning. Cambridge: MIT Press, 2016. 1–14.
- [42] Pan S, Hu R, Long G, Jiang J, Yao L, Zhang C. Adversarially regularized graph autoencoder for graph embedding. In: Rosenschein JS, ed. Proc. of the Int'l Joint Conf. on Artificial Intelligence. MA: Morgan Kaufmann Publishers, 2018.
- [43] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative adversarial networks. In: Lee DD, ed. Proc. of the Conf. on Neural Information Processing Systems (NIPS). Cambridge: MIT Press, 2014. 1–14.

- [44] Tu K, Cui P, Wang X, Yu PS, Zhu W. Deep recursive network embedding with regular equivalence. In: Krishnapuram B, ed. Proc. of the Int'l Conf. on Knowledge Discovery & Data Mining (KDD). New York: ACM, 2018. 2357–2366.
- [45] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*, 1997, 9(8): 1735–1780.
- [46] Yu W, Zheng C, Cheng W, Aggarwal CC, Song D, Zong B, Chen H, Wang W. Learning deep network representations with adversarially regularized autoencoders. In: Krishnapuram B, ed. Proc. of the Int'l Conf. on Knowledge Discovery & Data Mining (KDD). New York: ACM, 2018. 2663–2671.
- [47] Bruna J, Zaremba W, Szlam A, LeCun YJ. Spectral networks and locally connected networks on graphs. In: Sainath T, ed. Proc. of the Int'l Conf. on Learning Representation (ICLR). New York: OpenReview, 2014. 1–10.
- [48] Levie R, Monti F, Bresson X, Bronstein MM. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. *IEEE Trans. on Signal Processing*, 2018, 67(1): 97–109.
- [49] Li R, Wang S, Zhu F, Huang J. Adaptive graph convolutional neural networks. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2018. 1–10.
- [50] Micheli A. Neural network for graphs: A contextual constructive approach. *IEEE Trans. on Neural Networks*, 2009, 20(3): 498–511.
- [51] Duvenaud D, Maclaurin D, Aguilera-Iparraguirre J, Gómez-Bombarelli R, Hirzel T, Aspuru-Guzik A, Adams RPJ. Convolutional networks on graphs for learning molecular fingerprints. In: Lee DD, ed. Proc. of the Conf. on Neural Information Processing Systems (NIPS). Cambridge: MIT Press, 2015. 2224–2232.
- [52] Kearnes S, McCloskey K, Berndl M, Pande V, Riley P. Molecular graph convolutions: Moving beyond fingerprints. *Journal of Computer-aided Molecular Design*, 2016, 30(8): 595–608.
- [53] Li Q, Han Z, Wu XM. Deeper insights into graph convolutional networks for semi-supervised learning. In: Proc. of the AAAI Conf. on Artificial Intelligence. New York: AAAI, 2018. 1–14.
- [54] Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph attention networks. In: Sainath T, ed. Proc. of the Int'l Conf. on Learning Representations (ICLR). New York: Open Review, 2018. 1–10.
- [55] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In: Lee DD, ed. Proc. of the Conf. on Neural Information Processing Systems (NIPS). Cambridge: MIT Press, 2017. 1–10.
- [56] Xu K, Hu W, Leskovec J, Jegelka S. How powerful are graph neural networks? In: Sainath T, ed. Proc. of the Int'l Conf. on Learning Representations (ICLR). New York: Open Review, 2019. 1–10.
- [57] Qu Y, Bai T, Zhang W, Nie J, Tang J. An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation. In: Proc. of the 1st Int'l Workshop on Deep Learning Practice for High-dimensional Sparse Data. New York: ACM, 2019. 1–9.
- [58] Wang X, He X, Cao Y, Liu M, Chua TS. KGAT: Knowledge graph attention network for recommendation. In: Krishnapuram B, ed. Proc. of the Int'l Conf. on Knowledge Discovery & Data Mining (KDD). New York: ACM, 2019. 950–958.
- [59] Yang B, Yih WT, He X, Gao J, Deng L. Embedding entities and relations for learning and inference in knowledge bases. In: Sainath T, ed. Proc. of the Int'l Conf. on Learning Representations (ICLR). New York: Open Review, 2015. 1–10.
- [60] Zheng VW, Sha M, Li Y, Yang H, Fang Y, Zhang Z, Tan KL, Chang KCC. Heterogeneous embedding propagation for large-scale e-commerce user alignment. In: Aggarwal C, ed. Proc. of the 2018 IEEE Int'l Conf. on Data Mining (ICDM). Piscataway: IEEE, 2018. 1434–1439.
- [61] Hong H, Guo H, Lin Y, Yang X, Li Z, Ye J. An attention-based graph neural network for heterogeneous structural learning. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2020. 4132–4139.
- [62] Zhang C, Song D, Huang C, Swami A, Chawla NV. Heterogeneous graph neural network. In: Krishnapuram B, ed. Proc. of the Int'l Conf. on Knowledge Discovery & Data Mining (KDD). New York: ACM, 2019. 793–803.
- [63] Dettmers T, Minervini P, Stenetorp P, Riedel S. Convolutional 2D knowledge graph embeddings. In: Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2018. 1–10.
- [64] Zhao J, Wang X, Shi C, Hu B, Song G, Ye Y. Heterogeneous graph structure learning for graph neural networks. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence (AAAI). Menlo Park: AAAI, 2021. 1–12.

- [65] Jiang P, Han Y. Reasoning with heterogeneous graph alignment for video question answering. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2020. 11109–11116.
- [66] Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko O. Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS). 2013. 1–9.
- [67] Wang Z, Zhang J, Feng J, Chen Z. Knowledge graph embedding by translating on hyperplanes. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2014. 1112–1119.
- [68] Lin Y, Liu Z, Sun M, Liu Y, Zhu X. Learning entity and relation embeddings for knowledge graph completion. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2015. 2181–2187.
- [69] Nickel M, Tresp V, Kriegel HP. A three-way model for collective learning on multi-relational data. In: Jebara T, ed. Proc. of the ICML. New York: ACM, 2011. 809–816.
- [70] Kazemi SM, Poole DJ. Simple embedding for link prediction in knowledge graphs. In: Lee DD, ed. Neural Information Processing Systems (NIPS). Cambridge, 2018. 4289–4300.
- [71] Wang H, Zhao M, Xie X, Li W, Guo M. Knowledge graph convolutional networks for recommender systems. In: Liu L, ed. Proc. of the Web Conf. (WWW). New York: ACM, 2019. 3307–3313.
- [72] Zhu Z, Fan X, Chu X, Bi J. HGCN: A heterogeneous graph convolutional network-based deep learning model toward collective classification. In: Krishnapuram B, ed. Proc. of the Int'l Conf. on Knowledge Discovery & Data Mining (KDD). New York: ACM, 2020. 1161–1171.
- [73] Fu X, Zhang J, Meng Z, King I. MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In: Liu L, ed. Proc. of the Web Conf. (WWW). New York: ACM, 2020. 2331–2341.
- [74] Yun S, Jeong M, Kim R, Kang J, Kim HJJ. Graph transformer networks. In: Lee DD, ed. Proc. of the Conf. on Neural Information Processing Systems (NIPS). Cambridge: MIT Press, 2019. 11983–11993.
- [75] Hu B, Shi C, Zhao WX, Yu PS. Leveraging meta-path based context for top- n recommendation with a neural co-attention model. In: Krishnapuram B, ed. Proc. of the Int'l Conf. on Knowledge Discovery & Data Mining (KDD). New York: ACM, 2018. 1531–1540.
- [76] Sun Z, Yang J, Zhang J, Bozzon A, Huang LK, Xu C. Recurrent knowledge graph embedding for effective recommendation. In: Proc. of the ACM Conf. on Recommender Systems. New York: ACM, 2018. 297–305.
- [77] Dong Y, Chawla NV, Swami A. metapath2vec: Scalable representation learning for heterogeneous networks. In: Krishnapuram B, ed. Proc. of the Int'l Conf. on Knowledge Discovery and Data Mining (KDD). New York: ACM, 2017. 135–144.
- [78] Fu TY, Lee WC, Lei Z. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In: Winslett MS, ed. Proc. of the Conf. on Information and Knowledge Management (CIKM). New York: ACM, 2017. 1797–1806.
- [79] Wang X, Zhang Y, Shi C. Hyperbolic heterogeneous information network embedding. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI, 2019. 5337–5344.
- [80] Chen T, Sun Y. Task-guided and path-augmented heterogeneous network embedding for author identification. In: Carmel D, ed. Proc. of the ACM Int'l Conf. on Web Search and Data Mining. New York: ACM, 2017. 295–304.
- [81] Yu X, Ren X, Sun Y, Sturt B, Khandelwal U, Gu Q, Norick B, Han J. Recommendation in heterogeneous information networks with implicit user feedback. In: Proc. of the ACM Conf. on Recommender Systems. New York: ACM, 2013. 347–350.
- [82] Yu X, Ren X, Sun Y, Gu Q, Sturt B, Khandelwal U, Norick B, Han J. Personalized entity recommendation: A heterogeneous information network approach. In: Carmel D, ed. Proc. of the ACM Int'l Conf. on Web Search and Data Mining. New York: ACM, 2014. 283–292.
- [83] Zhang J, Shi X, Xie J, Ma H, King I, Yeung DYJ. GAAN: Gated attention networks for learning on large and spatiotemporal graphs. In: Proc. of the UAI. Arlington: AUAI, 2018. 339–349.
- [84] Seo Y, Defferrard M, Vandergheynst P, Bresson X. Structured sequence modeling with graph convolutional recurrent networks. In: Lee DD, ed. Proc. of the Int'l Conf. on Neural Information Processing. Berlin: Springer, 2018. 362–373.
- [85] Narayan A, Roe PHNJIP. Learning graph dynamics using deep neural networks. IFAC-PapersOnLine, 2018, 51(2): 433–438.
- [86] Niepert M, Ahmed M, Kutzkov K. Learning convolutional neural networks for graphs. In: Jebara T, ed. Proc. of the Int'l Conf. on Machine Learning (ICML). New York: ACM, 2016. 2014–2023.
- [87] Li Y, Yu R, Shahabi C, Liu YJ. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In: CoRR. 2017.

- [88] Goyal P, Kamra N, He X, Liu Y. Dyngem: Deep embedding method for dynamic graphs. In: Rosenschein JS, ed. Proc. of the IJCAI Int'l Workshop on Representation Learning for Graphs. Burlington: Morgan Kaufmann Publishers, 2017. 1–10.
- [89] Chen T, Goodfellow I, Shlens JJ. Net2net: Accelerating learning via knowledge transfer. In: Sainath T, ed. Proc. of the ICLR. New York: Open Review, 2016. 1–10.
- [90] Goyal P, Chhetri SR, Canedo A. dyngraph2vec: Capturing network dynamics using dynamic graph representation learning. Knowledge-based Systems, 2020, 187: 104816.
- [91] Rahman M, Al Hasan M. Link prediction in dynamic networks using graphlet. In: Proc. of the Joint European Conf. on Machine Learning and Knowledge Discovery in Databases. Berlin: Springer, 2016. 394–409.
- [92] Bonner S, Brennan J, Kureshi I, Theodoropoulos G, McGough AS, Obara B. Temporal graph offset reconstruction: Towards temporally robust graph representation learning. In: Proc. of the 2018 IEEE Int'l Conf. on Big Data (Big Data). IEEE, 2018. 3737–3746.
- [93] Wu T, Khan A, Gao H, Li CJ. Efficiently embedding dynamic knowledge graphs. In: CoRR. 2019. 1–14.
- [94] Ghosh P, Yao Y, Davis L, Divakaran A. Stacked spatio-temporal graph convolutional networks for action segmentation. In: Proc. of the IEEE/CVF Winter Conf. on Applications of Computer Vision. Piscataway: IEEE, 2020. 576–585.
- [95] Yu B, Yin H, Zhu ZJ. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In: Rosenschein JS, ed. Proc. of the 27th Int'l Joint Conf. on Artificial Intelligence Burlington: Morgan Kaufmann Publishers, 2018. 3634–3640.
- [96] Guo S, Lin Y, Feng N, Song C, Wan H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Palo Alto: AAAI Press, 2019. 922–929.
- [97] Heidari N, Iosifidis A. Progressive spatio-temporal graph convolutional network for skeleton-based human action recognition. In: Proc. of the ICASSP 2021-2021 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing (ICASSP). Piscataway: IEEE, 2021. 3220–3224.
- [98] Bolla MJDM. Spectra, euclidean representations and clusterings of hypergraphs. Discrete Mathematics, 1993, 117(1-3): 19–39.
- [99] Rodríguez JA. On the Laplacian eigenvalues and metric parameters of hypergraphs. Linear Multilinear Algebra, 2002, 50(1): 1–14.
- [100] Zhou D, Huang J, Schölkopf B. Beyond pairwise classification and clustering using hypergraphs. 2005. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.113.6516&rep=rep1&type=pdf>
- [101] Tan S, Guan Z, Cai D, Qin X, Bu J, Chen C. Mapping users across networks by manifold alignment on hypergraph. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI Press, 2014. 159–165.
- [102] Zhu Y, Guan Z, Tan S, Liu H, Cai D, He XJN. Heterogeneous hypergraph embedding for document recommendation. Neurocomputing, 2016, 216: 150–162.
- [103] Tan S, Bu J, Chen C, Xu B, Wang C, He X. Using rich social media information for music recommendation via hypergraph model. ACM Trans. on Multimedia Computing, Communications, 2011, 1–22.
- [104] Fatemi B, Taslakian P, Vazquez D, Poole DJ. Knowledge hypergraphs: Extending knowledge graphs beyond binary relations. arXiv: 1906.00137v1, 2019.
- [105] Hwang T, Tian Z, Kuangy R, Kocher JP. Learning on weighted hypergraphs to integrate protein interactions and gene expressions for cancer outcome prediction. In: Proc. of the IEEE Int'l Conf. on Data Mining. Piscataway: IEEE, 2008. 293–302.
- [106] Klamt S, Haus UU, Theis F. Hypergraphs and cellular networks. PLoS Computational Biology, 2009, 5(5): e1000385.
- [107] Huang Y, Liu Q, Zhang S, Metaxas DN. Image retrieval via probabilistic hypergraph ranking. In: Proc. of the 2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2010. 3376–3383.
- [108] Agarwal S, Branson K, Belongie S. Higher order learning with graphs. In: Jebara T, ed. Proc. of the 23rd Int'l Conf. on Machine Learning. New York: ACM, 2006. 17–24.
- [109] Shi J, Malik J. Normalized cuts and image segmentation. IEEE Trans. on Pattern Analysis Machine Intelligence, 2000, 22(8): 888–905.
- [110] Feng Y, You H, Zhang Z, Ji R, Gao Y. Hypergraph neural networks. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI Press, 2019. 3558–3565.

- [111] Jiang J, Wei Y, Feng Y, Cao J, Gao Y. Dynamic hypergraph neural networks. In: Rosenschein JS, ed. Proc. of the IJCAI. Burlington: Morgan Kaufmann Publishers, 2019. 2635–2641.
- [112] Chen H, Yin H, Sun X, Chen T, Gabrys B, Musial K. Multi-level graph convolutional networks for cross-platform anchor link prediction. In: Krishnapuram B, ed. Proc. of the 26th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. New York: ACM, 2020. 1503–1511.
- [113] Yi J, Park J. Hypergraph convolutional recurrent neural network. In: Krishnapuram B, ed. Proc. of the 26th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. New York: ACM, 2020. 3366–3376.
- [114] Chan THH, Liang Z. Generalizing the hypergraph laplacian via a diffusion process with mediators. *Theoretical Computer Science*, 2020, 806: 416–428.
- [115] Yadati N, Nitin V, Nimishakavi M, Yadav P, Louis A, Talukdar P. NHP: Neural hypergraph link prediction. In: Winslett MS, ed. Proc. of the 29th ACM Int'l Conf. on Information & Knowledge Management. New York: ACM, 2020. 1705–1714.
- [116] Scheinerman ER, Ullman DH. *Fractional Graph Theory: A Rational Approach to the Theory of Graphs*. Courier Corporation, 2011.
- [117] Yang C, Wang R, Yao S, Abdelzaher T. Hypergraph learning with line expansion. arXiv preprint arXiv: 04843, 2020.
- [118] Ding K, Wang J, Li J, Li D, Liu HJ. Be More with less: Hypergraph attention networks for inductive text classification. In: Proc. of the EMNLP. Stroudsburg PA: ACL, 2020. 4927–4936.
- [119] Zhang R, Zou Y, Ma JJ. Hyper-SAGNN: A self-attention based graph neural network for hypergraphs. In: Proc. of the Int'l Conf. on Learning Representations (ICLR). Open Review, 2019. 1–10.
- [120] Liu Z, Chen C, Yang X, Zhou J, Li X, Song L. Heterogeneous graph neural networks for malicious account detection. In: Proc. of the 27th ACM Int'l Conf. on Information and Knowledge Management. New York: ACM, 2018. 2077–2085.
- [121] Chen F, Gao Y, Cao D, Ji R. Multimodal hypergraph learning for microblog sentiment prediction. In: Proc. of the 2015 IEEE Int'l Conf. on Multimedia and Expo (ICME). Piscataway: IEEE, 2015. 1–6.
- [122] Gao Y, Wang M, Tao D, Ji R, Dai Q. 3-D object retrieval and recognition with hypergraph analysis. *IEEE Trans. on Image Processing*, 2012, 21(9): 4290–4303.
- [123] Huang X, Fang Q, Qian S, Sang J, Li Y, Xu C. Explainable interaction-driven user modeling over knowledge graph for sequential recommendation. In: Proc. of the 27th ACM Int'l Conf. on Multimedia. New York: ACM, 2019. 548–556.
- [124] Ma W, Zhang M, Cao Y, Jin W, Wang C, Liu Y, Ma S, Ren X. Jointly learning explainable rules for recommendation with knowledge graph. In: Liu L, ed. Proc. of the World Wide Web Conf. New York: ACM, 2019. 1210–1221.
- [125] Liu Z, Chen C, Li L, Zhou J, Li X, Song L, Qi Y. Geniepath: Graph neural networks with adaptive receptive paths. In: Zilberstein S, ed. Proc. of the AAAI Conf. on Artificial Intelligence. Menlo Park: AAAI Press, 2019. 4424–4431.
- [126] Bahnsen AC, Stojanovic A, Aouada D, Ottersten B. Cost sensitive credit card fraud detection using Bayes minimum risk. In: Proc. of the 12th Int'l Conf. on Machine Learning and Applications. Piscataway: IEEE, 2013. 333–338.
- [127] Hooi B, Shah N, Beutel A, Günnemann S, Akoglu L, Kumar M, Makhija D, Faloutsos C. Birdnest: Bayesian inference for ratings-fraud detection. In: Proc. of the 2016 SIAM Int'l Conf. on Data Mining. Philadelphia: SIAM, 2016. 495–503.
- [128] Jia Z, Abujabal A, Saha RR, Strötgen J, Weikum G. Tequila: Temporal question answering over knowledge bases. In: Proc. of the 27th ACM Int'l Conf. on Information and Knowledge Management. New York: ACM, 2018. 1807–1810.
- [129] Teney D, Liu L, van Den Hengel A. Graph-structured representations for visual question answering. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2017. 1–9.
- [130] Liang Z, Yang M, Deng L, Wang C, Wang B. Hierarchical depthwise graph convolutional neural network for 3D semantic segmentation of point clouds. In: Proc. of the 2019 Int'l Conf. on Robotics and Automation (ICRA). Piscataway: IEEE, 2019. 8152–8158.
- [131] Gao D, Li K, Wang R, Shan S, Chen X. Multi-modal graph neural network for joint reasoning on vision and scene text. In: Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition. IEEE, 2020. 12746–12756.
- [132] Te G, Hu W, Zheng A, Guo Z. RGCNN: Regularized graph CNN for point cloud segmentation. In: Proc. of the 26th ACM Int'l Conf. on Multimedia. New York: ACM, 2018. 746–754.
- [133] Landrieu L, Simonovsky M. Large-scale point cloud semantic segmentation with superpoint graphs. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2018. 4558–4567.

- [134] Chen J, Zhang J, Xu X, Fu C, Zhang D, Zhang Q, Xuan Q. E-LSTM-D: A deep learning framework for dynamic network link prediction. *IEEE Trans. on Systems, Man, Cybernetics: Systems*, 2021, 51(6): 3699–3712.
- [135] Lei K, Qin M, Bai B, Zhang G, Yang M. GCN-gan: A non-linear temporal link prediction model for weighted dynamic networks. In: *Proc. of the IEEE INFOCOM 2019-IEEE Conf. on Computer Communications*. Piscataway: IEEE, 2019. 388–396.
- [136] Yang M, Liu J, Chen L, Zhao Z, Chen X, Shen Y. An advanced deep generative framework for temporal link prediction in dynamic networks. *IEEE Trans. on Cybernetics*, 2019, 50(12): 4946–4957.
- [137] Ahmed NM, Chen L. An efficient algorithm for link prediction in temporal uncertain social networks. *Information Sciences*, 2016, 331: 120–136.
- [138] Divakaran A, Mohan A. Temporal link prediction: A survey. *New Generation Computing*, 2019, 38(1): 213–258.
- [139] Yu L, Du B, Hu X, Sun L, Han L, Lv W. Deep spatio-temporal graph convolutional network for traffic accident prediction. *Neurocomputing*, 2021, 423: 135–147.
- [140] Xiao G, Wang R, Zhang C, Ni A. Demand prediction for a public bike sharing program based on spatio-temporal graph convolutional networks. *Multimedia Tools Applications*, 2020, 80(15): 22907–22925.
- [141] Dai R, Xu S, Gu Q, Ji C, Liu K. Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data. In: Krishnapuram B, ed. *Proc. of the 26th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining*. New York: ACM, 2020. 3074–3082.
- [142] Li B, Li X, Zhang Z, Wu F. Spatio-temporal graph routing for skeleton-based action recognition. In: Zilberstein S, ed. *Proc. of the AAAI Conf. on Artificial Intelligence*. Menlo Park: AAAI Press, 2019. 8561–8568.
- [143] Konstantinova EV, Skorobogatov VA. Application of hypergraph theory in chemistry. *Discrete Mathematics*, 2001, 235(1–3): 365–383.
- [144] Zhang Z, Lin H, Zhao X, Ji R, Gao Y. Inductive multi-hypergraph learning and its application on view-based 3D object classification. *IEEE Trans. on Image Processing*, 2018, 27(12): 5957–5968.
- [145] Kim ES, Kang WY, On KW, Heo YJ, Zhang BT. Hypergraph attention networks for multimodal learning. In: *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*. Piscataway: IEEE, 2020. 14581–14590.
- [146] Zhang S, Cui S, Ding Z. Hypergraph spectral analysis and processing in 3D point cloud. *IEEE Trans. on Image Processing*, 2020, 30: 1193–1206.
- [147] Shao W, Peng Y, Zu C, Wang M, Zhang D. Hypergraph based multi-task feature selection for multimodal classification of Alzheimer's disease. *Computerized Medical Imaging Graphics*, 2020, 80: 101663.
- [148] Carlson A, Betteridge J, Kisiel B, Settles B, Hruschka E, Mitchell T. Toward an architecture for never-ending language learning. In: Zilberstein S, ed. *Proc. of the AAAI Conf. on Artificial Intelligence*. New York: AAAI, 2010. 1–12.
- [149] Dong X, Gabrilovich E, Heitz G, Horn W, Lao N, Murphy K, Strohmman T, Sun S, Zhang W. Knowledge vault: A Web-scale approach to probabilistic knowledge fusion. In: Krishnapuram B, ed. *Proc. of the 20th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*. New York: ACM, 2014. 601–610.
- [150] Tang X, Yuan R, Li Q, Wang T, Yang H, Cai Y, Song H. Timespan-aware dynamic knowledge graph embedding by incorporating temporal evolution. *IEEE Access*, 2020, 8: 6849–6860.
- [151] Fatemi B, Taslakian P, Vazquez D, Poole DJ. Knowledge hypergraphs: Extending knowledge graphs beyond binary relations. In: *CoRR*. 2019. 1–14.
- [152] <https://dblp.uni-trier.de>
- [153] <http://dl.acm.org/>
- [154] <https://ericdongyx.github.io/metapath2vec/m2v.html>
- [155] Yang Z, Cohen W, Salakhudinov R. Revisiting semi-supervised learning with graph embeddings. In: *Proc. of the Int'l Conf. on Machine Learning*. PMLR, 2016. 40–48.
- [156] Goldberg Y, Levy O. word2vec Explained: deriving Mikolov *et al.*'s negative-sampling word-embedding method. *arXiv preprint arXiv: 02263*, 2014.
- [157] <https://www.imdb.com/>
- [158] <https://grouplens.org/datasets/movielens/10m/>

- [159] <http://socialcomputing.asu.edu>
- [160] <https://www.yelp.com/dataset/challenge>
- [161] <https://www.ncbi.nlm.nih.gov/pubmed/>
- [162] <http://www.freebase.com/>
- [163] Socher R, Chen D, Manning CD, Ng A. Reasoning with neural tensor networks for knowledge base completion. In: Lee DD, ed. Proc. of the NIPS. Cambridge: MIT Press, 2013. 926–934.
- [164] Bordes A, Usunier N, Garcia-Duran A, Weston J, Yakhnenko OJA. Translating embeddings for modeling multi-relational data. In: Lee DD, ed. Proc. of the NIPS. Cambridge: MIT Press, 2013. 2787–2795.
- [165] Vrandečić D, Krötzsch MJCA. Wikidata: A free collaborative knowledgebase. Communications of the ACM, 2014, 57(10): 78–85.
- [166] Suchanek FM, Kasneci G, Weikum G. Yago: A core of semantic knowledge. In: Proc. of the 16th Int'l Conf. on World Wide Web. New York: ACM, 2007. 697–706.
- [167] <http://realitycommons.media.mit.edu/socialevolution.html>
- [168] Trivedi R, Farajtabar M, Biswal P, Zha H. Dyrep: Learning representations over dynamic graphs. In: Sainath T, ed. Proc. of the Int'l Conf. on Learning Representations. New York: Open Review, 2019.
- [169] <http://konect.uni-koblenz.de/networks/opsahl-ucsocial>
- [170] Chen C, Petty K, Skabardonis A, Varaiya P, Jia Z. Freeway performance measurement system: mining loop detector data. Transportation Research Record, 2001, 1748(1): 96–102.
- [171] Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, Viola F, Green T, Back T, Natsev P. The kinetics human action video dataset. arXiv preprint arXiv: 06950, 2017.
- [172] <https://archive.ics.uci.edu/ml/datasets/Yeast>
- [173] Chen DY, Tian XP, Shen YT, Ouhyoung M. On visual similarity based 3D model retrieval. In: Computer Graphics Forum. Wiley Online Library, 2003. 223–232.
- [174] Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J. 3D shapenets: A deep representation for volumetric shapes. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2015. 1912–1920.
- [175] <https://grouplens.org/datasets/hetrec-2011/>
- [176] <http://jmcauley.ucsd.edu/data/amazon>
- [177] <http://snap.stanford.edu/data/soc-RedditHyperlinks.html>
- [178] Chiang WL, Liu X, Si S, Li Y, Bengio S, Hsieh CJ. Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks. In: Krishnapuram B, ed. Proc. of the Int'l Conf. on Knowledge Discovery & Data Mining (KDD). New York: ACM, 2019. 257–266.
- [179] Yuan H, Yu H, Gui S, Ji S. Explainability in graph neural networks: A taxonomic survey. arXiv preprint arXiv: 15445, 2020.



刘杰(1997—), 男, 博士生, 主要研究领域为图表示学习, 图神经网络, 金融大数据.



尚学群(1973—), 女, 博士, 教授, CCF 高级会员, 美国 ACM 会员, 兼任中国计算机学会数据库专业委员会委员, 主要研究领域为数据挖掘, 机器学习, 生物信息学, 大数据.



宋凌云(1988—), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为计算机视觉, 基于金融大数据分析的风险预测, 基于教育大数据的认知诊断.



谭亚聪(1998—), 女, 硕士生, 主要研究领域为图神经网络, 知识图谱, 金融大数据.