中国科学: 信息科学 2024 年 第 54 卷 第 4 期: 853–871
SCIENTIA SINICA Informationis
论文

面向大规模数据的高效超图神经网络

吉书仪[1,3,4,5], 魏宇轩[1,3,4,5], 戴琼海[2,3,4,5], 高跃[1,3,4,5*]

1. 清华大学软件学院, 北京 100084
2. 清华大学自动化系, 北京 100084
3. 脑与认知智能北京实验室, 北京 100084
4. 北京信息科学与技术国家研究中心, 北京 100084
5. 清华大学脑与认知科学研究院, 北京 100084
* 通信作者. E-mail: gaoyue@tsinghua.edu.cn

收稿日期: 2022–09–30; 修回日期: 2023–04–05; 接受日期: 2023–08–16; 网络出版日期: 2024–04–08

国家自然科学基金 (批准号: 62021002, 62088102)、清华大学自主科研计划 (批准号: 20227020007)、北京市自然科学基金 (批准号: 4222025) 和之江实验室开放课题 (批准号: 2021KG0AB05) 资助项目

摘要 高阶关联广泛存在于现实世界中, 如社交网络、生物网络、交通网络等, 建模及优化高阶关联对于网络属性研究和演化趋势预测具有重要意义. 超图是一种灵活的数据结构, 能够自然地建模高阶关联. 近年来, 随着深度学习的发展, 基于超图建模的超图神经网络被广泛应用于面向高阶关联的表示学习. 然而, 现有的超图神经网络均基于直推学习范式, 虽然在小规模超图数据集上取得了不错的效果, 但难以应用到大规模数据上, 限制了其应用范围. 本文首先分析了现有超图神经网络方法在大规模数据上应用的挑战, 然后针对该问题提出了面向大规模数据的高效超图神经网络方法 (efficient hypergraph neural network, EHGNN). 针对现有方法空间、时间复杂度过高的问题, EHGNN 分别设计了超图采样模块和基于单阶段超图卷积的计算加速模块, 同时降低了超图神经网络的空间开销和时间开销, 使得超图神经网络适用于大规模超图数据, 显著增强了可扩展性. 在 4 个真实超图数据集上的实验结果验证了 EHGNN 的有效性和高效性.

关键词 超图计算, 超图神经网络, 高阶关联, 大规模数据, 节点分类

1 引言

现实世界由多个自然或人工的系统构成, 系统的组成要素相互联系、相互作用. 对系统进行建模、研究系统内部的相互关联有助于认识系统特性和预测演化趋势 [1]. 图是一种被广泛用于建模系统内关联的数据结构, 系统中的组成部分和相互作用被分别抽象成图中的节点和边 [2]. 然而, 图结构只能建模对象间的成对关联, 难以建模同时关联多个对象的相互作用, 即高阶关联. 事实上, 高阶关联普遍

引用格式: 吉书仪, 魏宇轩, 戴琼海, 等. 面向大规模数据的高效超图神经网络. 中国科学: 信息科学, 2024, 54: 853–871, doi: 10.1360/SSI-2022-0379
Ji S Y, Wei Y X, Dai Q H, et al. Efficient hypergraph neural network on million-level data (in Chinese). Sci Sin Inform, 2024, 54: 853–871, doi: 10.1360/SSI-2022-0379

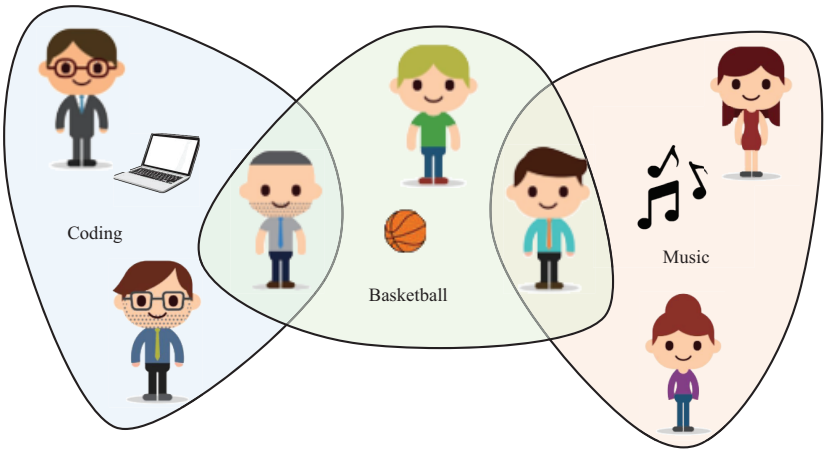

图 1　(网络版彩图) 社交网络中的高阶关联例子

Figure 1　(Color online) Example of high-order correlations in social network

存在于现实世界中. 例如, 在社交网络中, 多个用户可能共同形成一个兴趣小组; 在学术网络中, 一篇论文可能同时有多个合著者; 在生态网络中, 多个物种间能够相互调节、共存或竞争; 在药物 – 靶标网络中, 一种药物可能会同时与多个标靶之间产生相互作用; 在电商购物场景下, 一名顾客可能会购买多种商品, 而某件商品也可能会被多名顾客购买. 图 1 给出了一个社交网络中的高阶关联例子, 图中多个用户有共同的兴趣爱好, 因而形成了一个兴趣小组, 而简单图结构只能描述二元关联, 难以建模此类高阶复杂关联.

已有研究通过超图来建模此类复杂的高阶关联 [3]. 超图是一种灵活、高度可扩展的数据结构, 其灵活性主要来自于超边的自由度. 在简单图中, 一条边仅能连接两个节点; 而在超图中, 一条超边可以连接任意多数目的节点. 因此, 超图天然地具有建模多个要素间关联的能力. 在此基础上, 基于超图结构进行节点表示学习能够进一步分析及挖掘图结构数据中潜在的高阶复杂关联, 进而支撑下游任务, 例如推荐系统 [4]、药物发现 [5] 等.

随着深度学习的发展, 基于神经网络的超图表示学习方法受到了广泛关注. 一个典型的例子是 Feng 等 [6] 提出的谱域超图神经网络框架 (hypergraph neural network, HGNN). HGNN 结合超图建模与深度学习强大的表示能力, 首次引入了超图卷积、迭代学习数据表示. 在 HGNN 的基础上, Feng 等进一步提出了基于超图空域卷积的超图神经网络框架 HGNN$^+$ [7], 建立了更一般的两阶段超图卷积范式. Bai 等 [8] 提出了超图注意力神经网络 (hypergraph attention network, Hyper-Atten), 该网络遵循 HGNN 中定义的超图卷积模式. 受图注意力网络 (graph attention network, GAT) [9] 的启发, Hyper-Atten 引入了一个超边节点注意力学习模块, 自适应地识别同一超边中不同节点的重要性, 从而揭示节点之间的内在相关性. 此外, Yadati 等 [10] 提出了超图卷积网络, 一种基于超图的半监督学习方法 (hypergraph convolution network, HyperGCN). 基于超图的谱理论, 给定一个超图, HyperGCN 首先通过特定的策略将其转换成一个简单的加权图, 然后对该图执行标准的图卷积来学习数据表示.

上述超图神经网络方法在小规模数据集上取得了显著的提升, 但由于其均基于直推学习范式, 在应用于大规模数据时受到一定限制, 主要原因有二: 在空间消耗方面, 直推式超图神经网络在训练和测试时均需要进行整图计算, 其空间复杂度正比于超边数量, 因此当所需处理数据规模增大时, 所构造超图的超边数量急剧增加, 通用显存硬件设备难以满足模型所需空间; 在时间消耗方面, 现有的直推式超图神经网络方法基于 “节点 – 超边 – 节点” 的两阶段消息传递模式进行卷积, 计算复杂度较高, 运

算效率较低. 随着信息技术的发展, 现实世界中的超图数据规模越来越大, 可能包含百万甚至千万数量级的节点 [11]. 虽然目前已有一些面向大规模图数据设计的图神经网络方法, 例如, GraphSAGE [12] 通过迭代聚合节点的邻域信息来学习节点表示, Cluster-GCN [13] 将输入大规模图聚类成多个小图处理等, 但是还缺乏针对大规模超图数据设计的超图神经网络方法. 在这种背景下, 如何在时间 – 空间开销允许的范围内应用超图神经网络成为亟待解决的关键问题.

针对上述问题, 本文提出了一种面向大规模数据的高效超图神经网络方法 EHGNN (efficient hypergraph neural network). 该方法针对直推式超图神经网络空间复杂度和时间复杂度过高的问题, 分别设计了超图采样模块和基于单阶段卷积的计算加速模块. 其中, 超图采样模块包含分层采样算法和子超图预采样算法两种不同的采样方法, 将从大规模数据中直接构造的大超图分解为多个空间复杂度可控的子超图; 基于单阶段卷积的计算加速模块提出了 “节点 – 节点” 的超图卷积范式, 无须额外更新超边特征, 提高了计算效率. 在 4 个真实世界超图数据集上的实验结果表明, EHGNN 不仅增加了算法适用的数据规模, 也提升了超图卷积的运算效率, 显著增强了超图神经网络的可扩展性.

本文的其余部分组织如下: 第 2 节回顾了相关工作; 第 3 节进一步详细地阐述了所提出的面向大规模数据的超图神经网络模型 EHGNN; 第 4 节介绍了 EHGNN 在几个真实超图结构数据集上的实验结果, 并对结果进行了分析; 第 5 节对本文进行了总结, 并对值得探索的未来研究方向进行了讨论.

2 相关工作

近年来, 国内外学者针对超图学习方法的理论和应用进行了大量研究. 现有的超图学习方法主要可分为两类: 传统超图学习方法和基于深度学习的超图神经网络方法. 传统超图学习的核心是将超图嵌入问题建模为超图结构优化问题, 其概念由 Zhou 等 [14] 在 2006 年首次引入. Zhou 等基于超图模型的标签平滑假设, 将无向图分析中的谱聚类方法推广到超图上, 提出了基于谱的超图学习方法 (spectral-based hypergraph learning, SHL).

神经网络模型近来受到了学术界和工业界的广泛关注, 具有表达能力强、可扩展性高、自动化程度高的优势, 在目标检测 [15]、交通预测 [16] 及知识迁移 [17] 等领域得到了广泛的应用. 受此启发, 许多研究者对结合神经网络与超图建模优势的超图神经网络进行了大量研究. 一部分研究工作主要关注超图神经网络的模型设计. Feng 等 [6] 首次提出了超图神经网络模型 HGNN, 基于超图拉普拉斯矩阵 (Laplacian matrix) 定义了超图上的谱卷积. HGNN 中的超图谱卷积是两阶段卷积, 其对节点特征执行特征变换后进行 “节点 – 超边 – 节点” 的两阶段消息传递. 该框架被应用于引用网络节点分类与立体视觉对象识别两个任务, 并取得了优越的性能. 受 HGNN [6] 和 GAT [9] 的启发, Bai 等 [8] 进一步探索了端到端的超图注意力网络 Hyper-Atten, 将注意力机制引入超图卷积, 以学习一个能更好地揭示节点之间内在关联的动态转移矩阵. Yadati 等 [10] 提出了超图卷积神经网络 HyperGCN. 其首先基于中介点将超图转化为等效的简单带权图, 然后在该等效图上执行图卷积以进行信息传递, 并应用于半监督学习任务. 在异构网络建模方面, Fan 等 [18] 提出了异构超图变分自编码器网络, 通过引入两类节点建立了异构网络到异构超图间的映射, 并通过变分推断来求解隐变量分布.

另一部分研究工作聚焦于超图神经网络的应用. 在推荐系统领域, Xia 等 [4] 将超图应用于协同过滤, 并基于超图增强的跨视图对比学习框架捕获用户之间复杂的高阶依赖关联; Yu 等 [19] 提出了自监督多通道超图神经网络并应用于社交推荐任务, 通过定义不同的三角模体来建模用户间的多种社交高阶关联, 并将自监督学习集成到超图卷积网络中, 最大化层级互信息来学习交互. 在生物医学领域, Di 等 [20] 将超图神经网络应用于全尺寸病理图像存活预测任务中, 基于超图建模全尺寸病理图像块的高

阶关联, 然后结合贝叶斯 (Bayes) 优化准则进行排序预测; Nguyen 等 [5] 基于超图神经网络将药物交互超图和药物特征编码到潜在空间, 结合超图上的随机块模型学习药物潜在特征和副作用的多种类型的组合以预测药物间相互作用的副作用. 在交通预测领域, Wang 等 [21] 利用超图的可扩展性来描述乘客的不同出行模式, 基于超图神经网络建模地铁轨道客流的时间特性和空间特性以进行地铁客流量预测. 此外, 还有一些研究工作并没有直接应用超图神经网络, 而是将传统的超图学习与卷积神经网络相结合并应用于下游任务. 例如, 针对传统卷积神经网络面对类内差异较大数据性能较差的问题, Jin 等 [22] 提出了超图引导的卷积流形网络 (hypergraph induced convolutional manifold network, HCMN). HCMN 通过在每个小批次中构造超图, 并进行传统超图学习来提高模型对于噪声的鲁棒性, 提升深度卷积神经网络在计算机视觉任务中的性能.

上述超图神经网络方法均为直推式超图神经网络, 虽然在小规模数据上取得了显著的提升, 但难以应用于大规模数据, 实际应用受限.

3 高效超图神经网络模型

本节将首先概述所提出的高效超图神经网络模型 EHGNN, 描述符号和问题定义, 然后介绍本文所提出的超图采样模块、基于单阶段超图卷积的计算加速模块和模型细节并进行复杂度分析.

3.1 模型概述

EHGNN 主要由超图采样模块和基于单阶段超图卷积的计算加速模块组成. 给定数据, EHGNN 首先从数据中构建超图结构并初始化节点嵌入. 表征超图结构的关联矩阵 $\boldsymbol{H}$ 和节点嵌入 $\boldsymbol{X}$ 将作为模型的输入. EHGNN 模型由超图采样模块和多个超图卷积层堆叠而成. 在超图采样模块中, 从原始构建的大超图中采样出当前批次的节点集合用于计算. 完成局部超图结构采样后, 计算 “节点 – 节点” 局部邻接关系并进行单阶段超图卷积, 其中, 邻域节点进行特征聚合, 并基于全连接层进行特征变换, 最后进行特征的更新. EHGNN 能够高效编码大规模数据的高阶关联, 所学习到的低维节点嵌入能够进一步用于节点分类、链路预测等下游任务. 图 2 展示了本文所提出的面向大规模数据的 EHGNN 框架.

3.2 符号和问题定义

本小节首先介绍超图的定义. 超图包含节点集合与超边集合, 通常被定义为 $\mathcal{G}=(\mathcal{V},\mathcal{E},\boldsymbol{W})$, 其中, $\mathcal{V}$ 为节点集合, $\mathcal{E}$ 为超边集合, $\boldsymbol{W}$ 为超边的权重矩阵. 超边可以视为超图中节点集合的子集. 在带权超图中, 每个超边 $e\in\mathcal{E}$ 都被赋以权值 $w(e)$, 表示该连接在超图中的重要性. 由此, 超边的权重矩阵可以定义为 $\mathrm{diag}(\boldsymbol{W})=[w(e_1),w(e_2),\ldots,w(e_{|\mathcal{E}|})]$, 矩阵中对角线的每一个元素代表对应超边的权重. 给定一个超图 $\mathcal{G}$, 超图的结构通常通过一个关联矩阵 $\boldsymbol{H}\in\{0,1\}^{|\mathcal{V}|\times|\mathcal{E}|}$ 来描述, 其中每一个索引 $\boldsymbol{H}(v,e)$ 的值表示节点 v 是否属于超边 e:

$$\boldsymbol{H}(v,e)=\begin{cases}1, \text{ if } v\in e,\\ 0, \text{ if } v\notin e.\end{cases} \tag{1}$$

实际上, 超图关联矩阵中的索引值不仅限于 0 或 1, 可以是 $[0,1]$ 区间中的任意实数值.

超图中另外一个重要的定义是节点和超边的度. 如前所述, 在简单图中, 边的度恒为 2, 而超图中的超边度则非常灵活, 没有限制. 超图 $\mathcal{G}$ 中超边的度 $\delta(e)$ 和节点的度 $d(v)$ 分别定义为 $\delta(e)=$

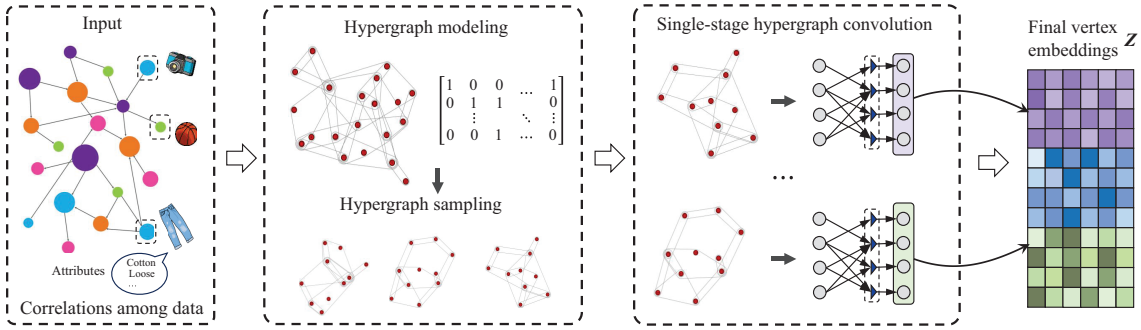

图 2　(网络版彩图) 面向大规模数据的高效超图神经网络框架图

Figure 2　(Color online) Overall pipeline of the proposed EHGNN

算法 1 Hypergraph neighborhood sampling algorithm

Input: Target vertex set $\mathcal{V}_t$; hypergraph incidence matrix $\boldsymbol{H}$; batch size B; neighborhood size W; adjacency function Adj; random sampling function RandomSample; matrix slice function Slice.

$\mathcal{E}_t = \mathrm{Adj}(\mathcal{V}_t, \boldsymbol{H})$;
$\mathcal{E}'_t = \mathrm{RandomSample}(\mathcal{E}_t, W)$;
$\mathcal{V}_b = \mathrm{Adj}(\mathcal{E}'_t, \boldsymbol{H}^{\mathrm{T}})$;
$\mathcal{V}'_b = \mathrm{RandomSample}(\mathcal{V}_b, W)$;
$\boldsymbol{H}_b = \mathrm{Slice}(\boldsymbol{H}, \boldsymbol{v}_b, \boldsymbol{h}_b)$;

Output: Batch vertex set $\boldsymbol{v}_b$; batch incidence matrix $\boldsymbol{H}_b$.

$\sum_{v\in\mathcal{V}} \boldsymbol{H}(v,e)$ 和 $d(v) = \sum_{e\in\mathcal{E}} \omega(e)\boldsymbol{H}(v,e)$. 据此, 可以进一步定义节点的度对角矩阵 $\boldsymbol{D}_v$ 和超边的度对角矩阵 $\boldsymbol{D}_e$, 矩阵中对角线的每一个元素代表对应节点和超边的度.

给定一个构建好的超图 $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$, 我们的目标是学习超图中节点的低维表示, 可以形式化地定义为学习映射函数 $f : (\mathcal{V}, \mathcal{E}, \boldsymbol{W}) \to \boldsymbol{Z}_{\mathcal{V}}$, 其中 $\boldsymbol{Z}_{\mathcal{V}} \in \mathbb{R}^{|\mathcal{V}|\times d}$ 是所学习到的节点低维表示.

3.3　超图采样模块

针对当前直推式超图神经网络采用整图计算方案导致空间复杂度过高从而不适用于大规模数据的问题, 本小节提出了超图采样模块, 并基于该模块采用分批计算方案来实现超图神经网络.

超图采样模块的核心思想是将从原始大规模数据中构造出的大超图分解为多个规模可控的子超图, 从而实现在可控空间复杂度下进行计算. 本小节进一步提出了两种超图采样方法: 分层采样和子超图预采样. 两种方法各有特色, 需根据不同的场景特点灵活选用.

在介绍分层采样和子超图预采样之前, 首先对节点邻域和超边邻域给出如下定义.

定义1 (节点邻域)　对于节点 v_i 而言, 其节点邻域定义为该节点所关联超边的集合: $\mathcal{N}_v(v_i) = \{e \mid v_i \in e\}$.

定义2 (超边邻域)　对于超边 e_i 而言, 其超边邻域定义为该超边所关联节点的集合: $\mathcal{N}_e(e_i) = \{v \mid v \in e_i\}$.

首先介绍超图邻域采样算法, 伪代码见算法 1. 给定第 t 批次的目标节点集合 $\mathcal{V}_t = \{v_i^t \mid i \in [1, B]\}$ (B 为批次规模, 即目标节点集合中节点的个数), 首先根据超图关联矩阵 $\boldsymbol{H}$ 计算每一个目标节点 v_t 的节点邻域 $\mathcal{N}_v(v_i^t)$, 即所关联的超边集合, 并取并集得到 $\mathcal{E}_t = \{\mathcal{N}_v(v_1^t) \cup \mathcal{N}_v(v_2^t) \cup \cdots \cup \mathcal{N}_v(v_n^t)\}$. 随后, 根据预设的批次规模超参数对该集合进行随机采样, 获得采样后的超边集合 $\mathcal{E}'_t = \{e_i^t \mid i \in [1, W]\}$, W 为

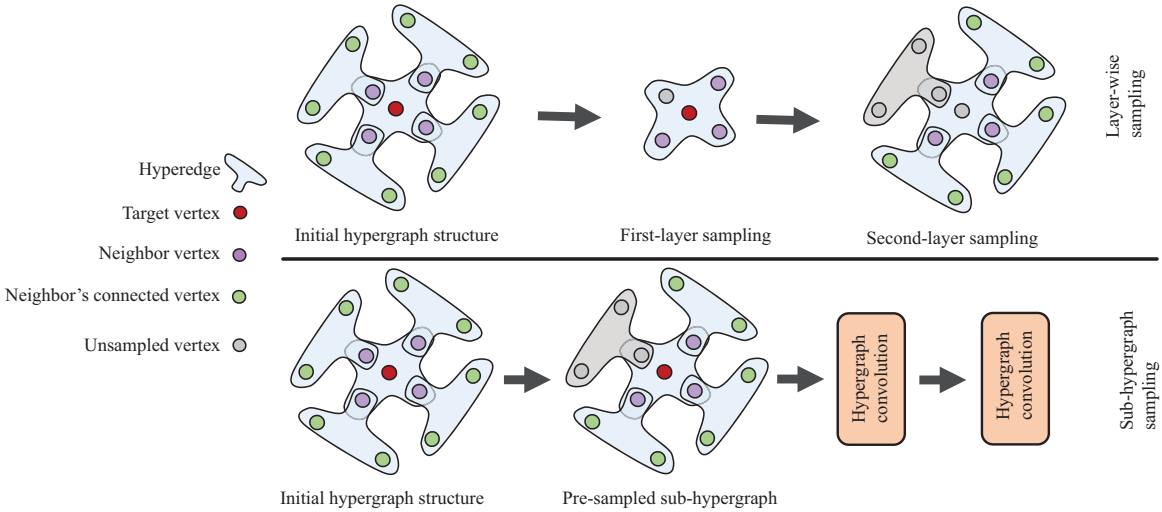

图 3 (网络版彩图) 分层采样与子超图预采样对比示意图

Figure 3 (Color online) Comparison of layer-wise sampling and sub-hypergraph sampling algorithms

算法 2 Hypergraph layer-wise sampling algorithm

Input: Hypergraph vertex set $\mathcal{V}$; hypergraph incidence matrix $\boldsymbol{H}$; batch size B; neighborhood size W; random sampling function RandomSample; hypergraph neighborhood sampling algorithm NeighborSample.

$\mathcal{V}_t = \text{RandomSample}(\mathcal{V}, B)$;

$\mathcal{V}_b', \boldsymbol{H}_b = \text{NeighborSample}(\mathcal{V}_t, \boldsymbol{H}, B, W)$;

Output: Target vertex set $\mathcal{V}_t$; batch vertex set $\mathcal{V}_b'$; batch incidence matrix $\boldsymbol{H}_b$.

邻域采样规模. 对该超边集合, 进一步对每条超边计算其超边邻域, 即其所关联节点的集合 $\mathcal{N}_e(e_i^b) = \{v \mid v \in e_i^t, e_i^t \in \mathcal{E}_t'\}$. 对于该批次所有目标节点, 该步骤获得节点集合 $\mathcal{V}_b$. 类似地, 对该集合根据预设的批次规模超参数进行随机采样, 获得该批次目标节点集计算所涉及的节点集合, 记为 $\mathcal{V}_b'$. 之后, 根据采样后的超边集合 $\mathcal{E}_t'$ 和节点集合 $\mathcal{V}_b'$ 对超图关联矩阵 $\boldsymbol{H}$ 进行矩阵切片, 获得当前批次采样的超图关联矩阵 $\boldsymbol{H}_b$, 在进行超图邻域采样时, 通过预先设定的批次规模超参数控制节点邻域和超边邻域的范围, 当所采样邻域范围超过该参数值时, 随机丢弃邻域内元素; 反之, 当所采样邻域范围小于该参数值时, 随机选择邻域元素进行复制来补齐.

接下来介绍分层采样和子超图预采样的具体细节. 分层采样和子超图预采样的对比如图 3 所示. 分层采样基于分层扩散的思想, 在超图神经网络的每个卷积层分别对当前超图进行采样, 得到多个子超图后再进行分批次的超图卷积. 详细描述见算法 2. 具体而言, 在进行分层采样时, 首先从节点集合 $\mathcal{V}$ 中随机采样出当前批次的目标节点集合 $\mathcal{V}_t$, 批次规模为 B, 而后对该集合通过超图邻域采样获得当次卷积计算所涉及的节点集合, 并通过从原超图关联矩阵和节点特征矩阵中进行切片获取对应的当前批次超图关联矩阵 $\boldsymbol{H}_b$ 和节点特征矩阵 $\boldsymbol{X}_b$. 分层采样通过控制每层交互的范围来实现数据分批计算及规模可控.

子超图预采样基于中心扩散的思想, 预先多层采样获得多个子超图, 再进行超图神经网络卷积计算. 详细描述见算法 3. 具体而言, 子超图预采样从每个目标节点出发, 逐次向外进行多层超图邻域采样以获取子超图结构, 生成当前批次所预采样涉及的节点集合 $\mathcal{V}_s$. 采样层数根据超图神经网络的深度确定. 对比而言, 分层采样每次需要存储当前层计算所涉及的所有参数, 而子超图预采样方法在开始

算法 3 Sub-hypergraph sampling algorithm

Input: Hypergraph vertex set $\mathcal{V}$; hypergraph incidence matrix $\boldsymbol{H}$; batch size B; neighborhood size W; the number of layers L; random sampling function RandomSample; hypergraph layer-wise sampling function LayerSample.

$\mathcal{V}_t = \text{RandomSample}(\mathcal{V}, B)$;
$\boldsymbol{H}_s = \Phi$;
$\mathcal{V}_b = \mathcal{V}_t$;
for $l = [1, \ldots, L]$ **do**
  $\mathcal{V}_{\text{new}}, \boldsymbol{H}_b = \text{NeighborSample}(\mathcal{V}_b, \boldsymbol{H}, B, W)$;
  $\mathcal{V}_b = \mathcal{V}_{\text{new}}$;
  $\mathcal{V}_s.\text{add}(\mathcal{V}_{\text{new}})$;
  $\boldsymbol{H}_s.\text{add}(\boldsymbol{H}_b)$;
end for

Output: Target vertex set $\mathcal{V}_t$; batch vertex set $\mathcal{V}_s$; batch incidence matrix $\boldsymbol{H}_s$.

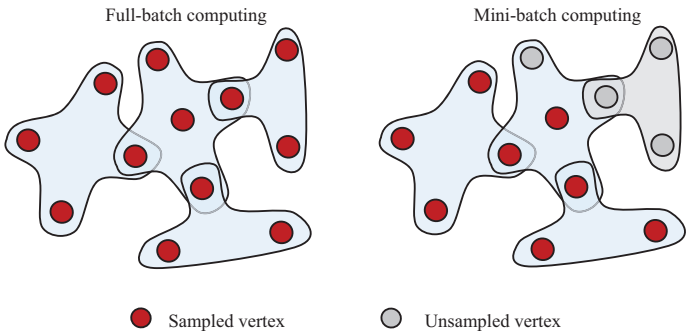

图 4 (网络版彩图) 整图计算方案与采样分批计算方案对比示意图

Figure 4 (Color online) Comparison of full-batch and mini-batch computing schema

计算前预先采样生成子超图. 因此, 在数据规模较大时, 设备显存可能难以满足分层采样的存储需求, 此时则应选择子超图预采样方法. 需要注意的是, 在测试阶段, 不需要存储训练阶段中可能产生的大量中间参数, 而分层采样方法相较子超图预采样方法能够使用更大的批次规模来分批计算, 从而提高时间效率.

进一步, 基于超图采样模块能够实现超图神经网络的分批计算, 整图计算方案和分批计算方案的对比如图 4 所示, 整图计算方案每次计算均需考虑所有的节点, 而分批计算方案仅需要采样部分节点进行计算. 相比于采用整图计算方案的直推式神经网络, 基于超图采样模块的超图神经网络, 能够将无法进行整图计算的大规模超图分解为多个可处理的小规模子超图, 从而实现大规模数据的分批计算. 该模型能够进一步支持归纳学习范式. 与直推学习相比, 归纳学习严格区分训练阶段和测试阶段, 训练集中不会包含测试集中的样本. 因此, 在应用于新数据时, 归纳学习无需重新训练整个模型, 显著提高了泛化性能. 基于超图采样模块进行分批计算能够在训练阶段仅计算训练样本节点, 而在未见节点上进行测试推理, 从而自然地遵循归纳学习的框架.

3.4 基于单阶段超图卷积的计算加速模块

针对现有的基于两阶段消息传递范式的超图神经网络方法时间复杂度较高的问题, 本小节提出了基于单阶段超图卷积的计算加速模块, 提高了超图神经网络的运算效率.

图 5 给出了两阶段超图卷积与单阶段超图卷积的对比示意图. 现有方法采用的两阶段消息传递

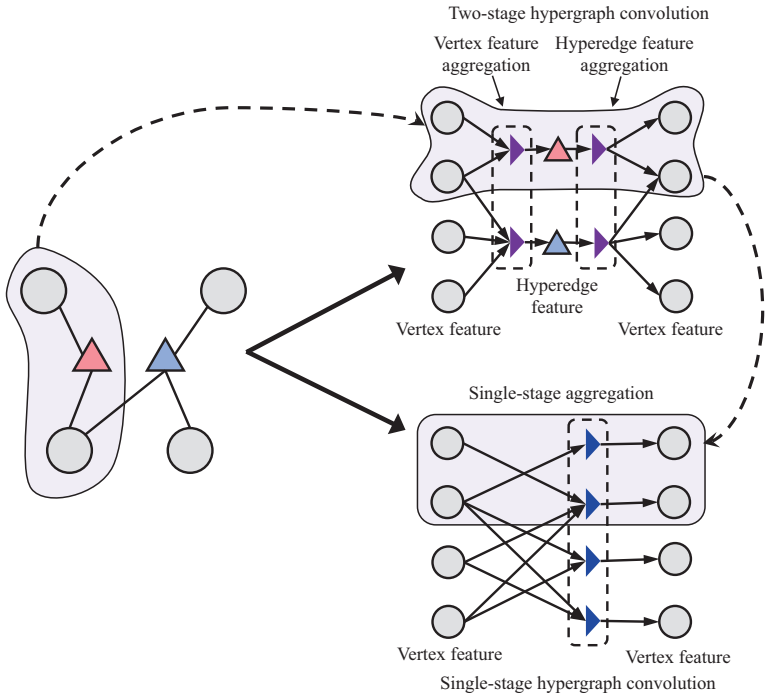

图 5　(网络版彩图) 单阶段超图卷积与两阶段超图卷积对比示意图

Figure 5　(Color online) Comparison of single-stage and two-stage hypergraph convolution

范式是在超图结构上进行 “节点 – 超边 – 节点” 的特征变换. 一次完整的超图卷积由一次节点卷积和一次超边卷积共同构成. 在第一阶段 (节点卷积阶段) 中, 节点的特征向所关联的超边聚合, 形成超边特征, 该过程由 $\boldsymbol{H}^{\mathrm{T}}$ 引导. 在第二阶段 (超边卷积阶段) 中, 上一阶段形成的超边特征向其所关联的节点聚合, 形成新的节点特征, 该过程由 $\boldsymbol{H}$ 引导. 相对仅进行 “节点 – 节点” 消息传递的图卷积操作而言, 两阶段超图卷积需要额外显式计算、更新超边特征, 带来了额外的计算复杂度, 时间开销较大.

为减小两阶段超图卷积的计算开销, 本小节提出了 “节点 – 节点” 单阶段超图卷积模块, 以避免超边特征的额外计算. 首先定义超图中的邻接节点.

定义3 (邻接节点)　在一个超图 $\mathcal{G}=(\mathcal{V},\mathcal{E},\boldsymbol{W})$ 中, 对于节点 v_i 与节点 v_j, 若存在 $v_i\in e$ 且 $v_j\in e$, $e\in\mathcal{E}$, 则称节点 v_i 与节点 v_j 互为邻接节点.

基于该定义, 在进行超图神经网络的分批计算时, 能够在当前批次节点集内获取节点间的局部邻接关系, 并进行从源节点特征到目标节点特征的特征聚合. 最后, 通过非线性激活函数对聚合后的特征施加非线性特征变换. 上述过程构成了一个完整的单阶段超图卷积层, 其不再需要显式计算超边特征, 而是将超边视为节点关联的中介, 基于超边来捕获节点的高阶邻接关系. 因此, 一次单阶段超图卷积只需要进行一次特征聚合, 从而显著降低了超图卷积的运算开销.

3.5　模型细节

在实际操作过程中, 模型的计算又进一步可分为训练和测试两个阶段, 由于不同阶段具有不同的特点, 因此在模型中超图采样方式的选择上又略有不同. 在模型的训练阶段, 需进行前向传播和反向传播. 反向传播过程需要保留大量的中间数据, 其空间消耗远高于前向传播过程, 约为 2∼4 倍 (与所选择的优化器有关). 因此, 在模型的训练阶段, 超图采样模块选择子超图预采样, 如算法 4 所示. 给

算法 4 EHGNN: training phase

Input: Vertex feature matrix $\boldsymbol{X} \in \mathbb{R}^{N_v \times N_c}$; training sample label vector $\boldsymbol{y} \in \mathbb{R}^{N_v}$; sparse hypergraph incidence matrix $\boldsymbol{H} \in \mathbb{R}^{2 \times N_i}$; batch size B; neighborhood size W; number of layers L; sub-hypergraph sampling function SubgraphSample; feature aggregation function Aggr; multi-layer perception function Fc; vector maximum function Argmax; loss function LossFn; back propagation function Backward.

for $\mathcal{V}_t, \mathcal{V}_s, \boldsymbol{A}_s = \text{SubgraphSample}(\boldsymbol{H})$ **do**
  for $l = [1, \ldots, L]$ **do**
    $\mathcal{V}_l = \mathcal{V}_s[l]$;
    $\boldsymbol{A}_l = \boldsymbol{A}_s[l]$;
    $\boldsymbol{X}_t = \text{Fc}^{(l)}\big(\text{Aggr}\left(\boldsymbol{X}[\mathcal{V}_t], \boldsymbol{X}[\mathcal{V}_l], \boldsymbol{A}_l\right)\big)$;
  end for
  $\boldsymbol{z}_t = \text{Argmax}(\boldsymbol{X}_t, \text{dim} = 1)$;
  $\text{loss} = \text{LossFn}(\boldsymbol{z}_t, \boldsymbol{y}[\mathcal{V}_t])$;
  Backward(loss);
end for

算法 5 EHGNN: testing phase

Input: Vertex feature matrix $\boldsymbol{X} \in \mathbb{R}^{N_v \times N_c}$; training sample label vector $\boldsymbol{y} \in \mathbb{R}^{N_v}$; sparse hypergraph incidence matrix $\boldsymbol{H} \in \mathbb{R}^{2 \times N_i}$; batch size B; neighborhood size W; number of layers L; hypergraph layer-wise sampling function LayerSample; feature aggregation function Aggr; multi-layer perception function Fc; vector maximum function Argmax; vertex concatenation function Concatenate.

for $l = [1, \ldots, L]$ **do**
  $\boldsymbol{X}_s = \Phi$;
  for $\mathcal{V}_t, \mathcal{V}_b, \boldsymbol{A}_b = \text{LayerSample}(\boldsymbol{H})$ **do**
    $\boldsymbol{X}_t = \text{Fc}^{(l)}\big(\text{Aggr}(\boldsymbol{X}[\mathcal{V}_t], \boldsymbol{X}[\mathcal{V}_b], \boldsymbol{A}_b)\big)$;
    $\boldsymbol{X}_s.\text{add}(\boldsymbol{X}_t)$;
  end for
  $\boldsymbol{X} = \text{Concatenate}(\boldsymbol{X}_s)$;
end for
$\boldsymbol{z} = \text{Argmax}(\boldsymbol{X}, \text{dim} = 1)$;

Output: Predicted result $\boldsymbol{z} \in \mathbb{R}^{N_v}$.

定节点的特征矩阵 $\boldsymbol{X}$、训练样本的标签向量 $\boldsymbol{y}$、稀疏表示下的超图关联矩阵 $\boldsymbol{H}$, 其预先采样出多个小批次的目标节点集合及其多层近邻节点, 构造出多个子超图, 并进行分批次的多层超图卷积. 所有批次的结果最终会被汇总以计算损失并回传更新参数. 相对训练阶段而言, 模型的测试阶段仅包含前向传播过程, 因此可选用分层采样并选择大批次规模, 从而避免频繁地从内存将数据中加载到显存中, 减小时间开销. 详细描述见算法 5.

3.6 复杂度分析

现有直推式超图神经网络在大规模数据上应用的主要瓶颈在于整图计算的空间开销. 所提出的基于超图采样模块的超图神经网络采用分批计算方案, 将无法进行整图计算的大规模超图分解为多个可处理的小规模子超图. 分批计算的批次规模可根据空间资源灵活设定, 与超图固有的属性 (如节点数和超边数) 无关. 由于神经网络的训练阶段包含前向计算、反向传播和优化等步骤, 其空间开销远大于测试阶段, 因此本小节以神经网络的训练阶段空间复杂度来分析基于超图采样模块的超图神经网络复杂度.

给定一个具有 N 个节点, M 条超边的超图 $\mathcal{G}$, 现有直推式超图神经网络直接对该超图进行整图计算, 其空间复杂度为 $O(NM)$, 正比于超图关联矩阵 $\boldsymbol{H}$. 假设根据存储资源设定基于超图采样模块的超图神经网络的采样批次规模为 B, 则对应的空间复杂度为 $O(B^2)$. 通常, 当应用于大规模数据时, B 远小于 N 和 M. 因此, 所提出的基于超图采样模块的超图神经网络能够有效降低超图计算的空间复杂度, 显著减小空间开销.

4 实验内容

本节对 EHGNN 模型在几个真实超图结构数据集上的有效性进行了实验验证. 同时, 进行了一系列消融实验, 以进一步分析 EHGNN 模型的性能.

4.1 节点分类实验

4.1.1 数据集

为全面评估不同方法的有效性, 我们在 4 个真实世界的超图数据集上进行了实验, 包含 2 个大规模超图数据集 (Amazon-reviews [23] 和 CIKM19-ECOMM [24]) 以及 2 个小规模超图数据集 (Walmart-trips [25] 和 House-bills [26]). 数据集的具体描述如下.

• **Amazon-reviews [23].** 该数据集描述了亚马逊在线平台上的用户对于不同商品的评论, 包含 2268231 个商品和 4285363 个用户, 商品属于 29 个不同的类别. 构建超图时, 将商品建模为节点, 用户建模为超边, 关联该用户评论过的节点.

• **CIKM19-ECOMM [24].** 该数据集源自 CIKM 2019 EComm AI: Efficient User Interests Retrieva 比赛, 描述了电商平台上的用户 – 商品交互, 包含 1000000 个用户和 3849407 个商品. 用户根据其购买力的不同共分为 9 种类别. 构建超图时, 将用户建模为节点, 商品建模为超边, 关联与该商品交互过的用户.

• **Walmart-trips [25].** 该数据集源自 Kaggle 比赛, 描述了沃尔玛在线电商网站上的购物行为关系, 包含 88860 个商品及 69906 条购买记录, 商品根据其种类共分为 11 个类别. 构建超图时, 将商品建模为节点, 购买记录建模为超边, 关联在单次购买记录中所有购买的商品.

• **House-bills [26].** 该数据集描述了美国国会议案的共同支持关系, 包含 1494 个国会议员和 60987 条议案. 国会议员根据其所属政党共分为 2 类. 构建超图时, 将议员建模为节点, 议案建模为超边, 关联该议案的发起人和所有共同支持者.

表 1 展示了 4 个数据集的统计信息. 对于所有实验, 训练集、测试集、验证集的比例划分均为 8:1:1. 对于基于超图的对比方法, 直接通过数据集介绍中的方式构造超图. 对于基于图的对比方法, 首先使用团扩展 [27] 将所构建的超图扩展为简单图, 再运行基于图的节点分类方法. 由表 1 可见, 团扩展后得到的简单图的边数量要远多于原超图中的超边数量.

4.1.2 基准方法

我们将所提出的 EHGNN 模型与以下 13 个基准方法进行对比, 包含传统图学习方法 (traditional graph learning)、图表示学习方法 (graph representation learning)、图神经网络方法 (graph neural networks)、传统超图学习方法 (traditional hypergraph learning) 以及超图神经网络方法 (hypergraph neural networks) 共 5 大类方法.

表 1 超图数据集信息统计

Table 1 Hypergraph dataset statistics

Dataset	#Nodes	#Edges	#Extended edges	#Classes
Amazon-reviews	2268231	4285363	6067093005	29
CIKM19-ECOMM	1000000	3849407	9380156656	9
Walmart-trips	88860	69906	4244974	11
House-bills	1494	60987	993658	2

• 传统图学习方法包含图最近邻算法 (nearest neighborhood, NN) [28] 和标签传播算法 (label propagation algorithm, LPA) [29]. 图最近邻算法基于投票思想, 对于一个待分类节点, 其类别标签由图上已知类别的近邻节点投票决定. LPA 是一种图上的半监督学习算法, 基于节点间的距离在节点间传播标签直至收敛. LPA 可通过解析解和迭代解两种形式来求解问题. 解析解求解的复杂度较高, 因此在小规模数据集上实验时, 采用 LPA 解析解形式; 在大规模数据集上实验时, 采用 LPA 迭代解形式求解.

• 图表示学习方法包括 DeepWalk [30] 和 Node2Vec [31] 方法. DeepWalk 主要基于随机游走方法采样获得节点序列, 并使用 skipgram 方法学习节点嵌入. 基于该嵌入, 使用多层神经网络来进行后续的节点分类. Node2Vec 在 DeepWalk 的基础上, 采用有偏随机游走的策略来获取节点序列并学习节点嵌入.

• 图神经网络方法包括 GraphSAGE [12], Cluster-GCN [13] 和 GraphSAINT [32]. 这 3 种方法均采用归纳学习范式. GraphSAGE 采用分批计算方案, 每次采样一个批次节点近邻来学习节点嵌入, 并学习特征聚合函数. Cluster-GCN 采用基于图聚类的分批计算方法, 首先在预处理过程中将图划分为密集连接的簇, 并随机选择一个或多个簇来构成当前批次进行训练. GraphSAINT 首先从原图中采样出子图, 并基于子图来训练图卷积网络, 所提出的正则化方法和采样策略能够消除方差并减少方差.

• 传统超图学习方法包括基于谱的超图学习方法 SHL [14] 和超图标签传播方法 (cross diffusion on multi-hypergraph, CDMH) [33]. SHL 将原本对无向图进行分析的谱聚类方法推广到超图上, 并在谱超图聚类方法的基础上进一步提出了超图嵌入和转导推理算法, 旨在最小化超图上具有较强连接的节点之间的标签差异. CDMH 通过在超图结构上进行标签扩散来更新标签投影矩阵, 直至该过程收敛.

• 超图神经网络方法包括 HGNN [6], HyperGCN [10], HNHN [34] 和 HGNN$^+$ [7] 方法, 这几种方法均采用直推学习范式. HGNN 首次提出了基于超图谱域卷积的超图神经网络框架, 引入了超图拉普拉斯矩阵来对节点特征进行平滑. HyperGCN 是基于超图的谱理论设计的. HyperGCN 首先通过特定的策略将给定超图转换成简单的加权图, 然后对该图采用标准 GCN 来学习数据表示. HNHN 提出在超图卷积网络中将非线性激活函数同时应用于节点和超边, 并采用了一种灵活的归一化方案, 根据数据集调整超边和顶点的重要性. HGNN$^+$ 在 HGNN 的基础上进一步提出了基于超图空域卷积的超图神经网络框架, 其引入超路径消息传递用于聚合超图中相邻节点信息.

若无特别标明, 则下文中提到的 EHGNN 模型表示使用超图采样模块的归纳式超图神经网络, EHGNN-fast 表示 EHGNN 的加速版本, 即在 EHGNN 的基础上加入 3.4 小节中基于单阶段超图卷积的计算加速模块.

4.1.3 评测指标

我们在超图数据节点分类任务上进行了实验, 采用 Accuracy, Precision, Recall, F1 score 和 AUC

表 2　超参数设置

Table 2　Hyperparamter settings

Dataset	#Sampled neighboring nodes	Embedding dimension	Hidden dimension	Learning rate
Amazon-reviews	32	64	16	0.001
CIKM19-ECOMM	64	64	16	0.001
Walmart-trips	64	1024	64	0.001
House-bills	64	16	16	0.08

等指标来评估性能. 首先介绍二分类任务中的指标定义.

• Accuracy = $\frac{\mathrm{TP}+\mathrm{TN}}{N}$, 其中 TP 表示预测正确的正样本, TN 表示预测正确的负样本, N 表示样本总量.

• Precision = $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$, 其中 FP 表示预测错误的正样本.

• Recall = $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$, 其中 FN 表示预测错误的负样本.

• F1 score = $\frac{2\times\mathrm{Precision}\times\mathrm{Recall}}{\mathrm{Precision}+\mathrm{Recall}}$.

• AUC = ROC 曲线下的面积.

对于多标签分类任务, 本文均采用 macro 指标计算: Accuracy 指标的计算方式不变, 对于其余指标, 将多分类问题转化为多个二分类问题, 分别计算每个二分类问题下的对应指标, 最后求平均值得到多标签任务中的 Precision, Recall, F1 score 和 AUC.

4.1.4　参数设置

对于所有数据集, 本文均采用两层 EHGNN 进行学习, 在第一层和第二次超图卷积层后分别接一层 ReLU 激活层和一层 logSoftmax 层. 不同数据集上的参数通过在验证集上进行超参数调优确定, 具体参数设置如表 2 所示. 采样批次规模均设定为 2048.

所有实验均在 12 核、640 G 内存的 CPU 服务器和 40 核、256 G 内存的 GPU 服务器上运行, GPU 服务器配置有显存为 32 G 的 Tesla V100 显卡. 传统图学习方法、传统超图学习方法及图表示学习的节点嵌入学习阶段无需使用显存计算, 使用前述 CPU 服务器进行计算; 图表示学习的分类阶段、图神经网络方法和超图神经网络方法需使用显存计算, 在前述 GPU 服务器上进行计算.

4.1.5　实验结果及分析

本文在 4 个真实世界超图数据集上对 EHGNN 和对比方法进行了实验. 在实验结果中, 用 “OOM” (“out of memory”) 表示该方法超过当前内存限制无法运行. 首先是两个大规模数据集 Amazon-reviews 和 CIKM19-ECOMM. 由表 1 可见, Amazon-reviews 和 CIKM19-ECOMM 数据集团扩展后的简单图中包含十亿数量级的边, 因此只有传统图学习方法 (NN, LPA) 和 EHGNN 模型能在两个大规模数据集上运行. DeepWalk 在 Amazon-reviews 数据集上可运行, 而在 CIKM19-ECOMM 数据集上会超出内存限制无法运行. Node2Vec 方法、图神经网络方法、传统超图学习方法以及超图神经网络方法运算均会超出 CPU 服务器的内存限制, 因而无法在这两个大规模超图数据集上运行.

在大规模超图数据集上的实验结果如表 3 和 4 所示. 从中可以观察到, 所提出的 EHGNN 模型在所有指标上均大幅超过对比方法. 例如, 在 Amazon-reviews 数据集中的 Accuracy 指标上, EHGNN 超过 DeepWalk 方法约 32 个百分点; 在 Precision 指标上, EHGNN 相对于 DeepWalk 性能提高了 41.71%. 性能提升主要有两方面原因: 一方面, 基于图的方法难以直接建模超图数据中对象间的高阶关联, 只

表 3 Amazon-reviews 数据集实验结果

Table 3 Experimental results on Amazon-reviews dataset[a)]

Method category	Method	Accuracy	Precision	Recall	F1 score	AUC
Traditional graph learning	NN [28]	0.4689±0.0000	0.6497±0.0000	0.2083±0.0000	0.2479±0.0000	0.4914±0.0000
	LPA [29]	0.4934±0.0000	0.6796±0.0000	0.1534±0.0000	0.1907±0.0000	0.4985±0.0000
Graph representation learning	DeepWalk [30]	0.6467±0.0018	0.4932±0.0036	0.3641±0.0031	0.3805±0.0036	0.4903±0.0004
Hypergraph neural networks	EHGNN	**0.9664±0.0005**	**0.9103±0.0068**	**0.9138±0.0042**	**0.8843±0.0088**	**0.9563±0.0021**

a) The best results are marked in bold.

表 4 CIKM19-ECOMM 数据集实验结果

Table 4 Experimental results on CIKM19-ECOMM dataset[a)]

Method category	Method	Accuracy	Precision	Recall	F1 score	AUC
Traditional graph learning	NN [28]	0.1939±0.0000	0.0991±0.0000	0.1309±0.0000	0.0608±0.0000	0.5016±0.0000
	LPA [29]	0.2770±0.0000	0.1101±0.0000	0.1112±0.0000	0.0483±0.0000	0.5000±0.0000
Graph representation learning	DeepWalk [30]	OOM	OOM	OOM	OOM	OOM
Hypergraph neural networks	EHGNN	**0.3308±0.0007**	**0.2491±0.0028**	**0.2267±0.0032**	**0.2034±0.0086**	**0.5633±0.0016**

a) The best results are marked in bold.

能使用超图数据的团扩展结果作为输入, 而超图的团扩展并不能严格等效原有的超图结构, 存在信息损失, 无法完全保留超图的结构信息, 而 EHGNN 基于超图建模, 能够捕获数据中对象的高阶关联并进行推理, 从而获得了性能提升; 另一方面, EHGNN 基于超图神经网络进行学习, 相对于传统方法具有更强的学习能力和表达能力.

我们还在两个小规模超图数据集 Walmart-trips 和 House-bills 上进行了实验. Walmart-trips 数据集中包含 8 万余个节点, 约 7 万条超边, 其团扩展后获得的简单图中包含四百万余条边. 在当前实验环境下, 基于直推学习范式的超图神经网络方法均无法在该数据集上运行. 在小规模超图数据集上的实验结果如表 5 和 6 所示.

从以上实验结果中, 可以得到如下的分析和结论:

• 基于神经网络的方法性能普遍优于传统方法 (包括传统图学习方法与传统超图学习方法). 这主要是因为深度学习强大的表达能力使得模型能够更好地建模节点的特性.

• 基于超图的方法性能普遍优于基于图的方法, 且 EHGNN 在两个数据集的所有指标上均取得了最优结果. 例如, 对于 House-bills 数据集, HGNN$^+$ 相对于 GraphSAGE 方法在 Recall 指标上取得了 25.79% 的性能提升. 特别地, 所提出的模型 EHGNN 相对于 GraphSAGE 在 Recall 指标上取得了 31.65% 的性能提升. 这主要是因为图结构由于其固有限制, 难以建模超图数据中节点间的高阶关联, 基于超图团扩展获得的简单图相对于原超图存在信息损失. 而 EHGNN 能够更好地建模数据间的高阶关联, 从而获得更优的性能.

4.1.6 算法运行时间和空间分析

为对比 EHGNN 模型和对比方法的时空运行效率, 本文在 4 个数据集上对 EHGNN 和对比方法的时间、空间开销进行了统计分析. 表 7 展示了在两个大规模数据集 Amazon-reviews 和 CIKM19-ECOMM 上 EHGNN 模型和对比方法的空间消耗情况. 从结果中可以看到, 除了会遇到内存/显存溢出问题而无法运行的 Node2Vec、传统超图学习方法、图/超图神经网络方法外, EHGNN 方法的空间消耗远远小于其他对比方法. 其主要原因有两方面: 首先, EHGNN 能够直接建模超图结构的数据, 避免使用超图团扩展生成数据; 另一方面, EHGNN 能够基于所提出的超图采样模块实现超图分批计算, 避免了整图计算的巨大空间开销.

表 8 展示了在两个大规模数据集 Amazon-reviews 和 CIKM19-ECOMM 上 EHGNN 模型和对比

表 5 Walmart-trips 数据集实验结果

Table 5 Experimental results on Walmart-trips dataset[a)]

Method category	Method	Accuracy	Precision	Recall	F1 score	AUC
Traditional graph learning	NN [28]	0.3054±0.0000	0.7311±0.0000	0.1937±0.0000	0.2195±0.0000	0.5619±0.0000
	LPA [29]	0.3945±0.0000	0.7293±0.0000	0.1501±0.0000	0.1435±0.0000	0.5362±0.0000
Graph representation learning	Node2Vec [31]	0.4593±0.0111	0.6262±0.0452	0.2377±0.0046	0.2576±0.0038	0.5357±0.0035
	DeepWalk [30]	0.7064±0.0031	0.6939±0.0406	0.4557±0.0047	0.4886±0.0058	0.6085±0.0408
Graph neural networks	GraphSAGE [12]	0.7513±0.0041	0.7501±0.0042	0.5433±0.0074	0.6012±0.0067	0.7571±0.0040
	Cluster-GCN [13]	0.7849±0.0032	0.7843±0.0070	0.4939±0.0064	0.5109±0.0073	0.7354±0.0479
	GraphSAINT [32]	0.5656±0.0077	0.5670±0.0071	0.3441±0.0159	0.3785±0.0195	0.6461±0.0084
Traditional hypergraph learning	SHL [14]	0.7205±0.0000	0.7228±0.0000	0.5164±0.0000	0.5789±0.0000	0.7415±0.0000
	CDMH [33]	0.6705±0.0000	0.7302±0.0000	0.4435±0.0000	0.5132±0.0000	0.7016±0.0000
Hypergraph neural networks	HGNN [6]	OOM	OOM	OOM	OOM	OOM
	HGNN+ [7]	OOM	OOM	OOM	OOM	OOM
	HyperGCN [10]	OOM	OOM	OOM	OOM	OOM
	HNHN [34]	OOM	OOM	OOM	OOM	OOM
	EHGNN	**0.7904±0.0023**	**0.7907±0.0066**	**0.6067±0.0076**	**0.6595±0.0065**	**0.7911±0.0039**

a) The best results are marked in bold.

表 6 House-bills 数据集实验结果

Table 6 Experimental results on House-bills dataset[a)]

Method category	Method	Accuracy	Precision	Recall	F1 score	AUC
Traditional graph learning	NN [28]	0.5034±0.0000	0.5581±0.0000	0.5259±0.0000	0.4374±0.0000	0.5259±0.0000
	LPA [29]	0.6242±0.0000	0.7926±0.0000	0.6000±0.0000	0.5358±0.0000	0.6000±0.0000
Graph representation learning	Node2Vec [31]	0.5436±0.0134	0.5379±0.0182	0.5289±0.0146	0.5062±0.0221	0.5289±0.0146
	DeepWalk [30]	0.6779±0.0067	0.7214±0.0106	0.6628±0.0072	0.6486±0.0103	0.6628±0.0072
Graph neural networks	GraphSAGE [12]	0.6644±0.0427	0.7101±0.0560	0.6486±0.0563	0.6307±0.0632	0.6486±0.0563
	Cluster-GCN [13]	0.6040±0.0212	0.6123±0.0323	0.6095±0.0198	0.6029±0.0461	0.6095±0.0198
	GraphSAINT [32]	0.6174±0.0341	0.6332±0.0382	0.6254±0.0361	0.6140±0.0338	0.6254±0.0361
Traditional hypergraph learning	SHL [14]	0.8792±0.0000	0.9072±0.0000	0.8714±0.0000	0.8751±0.0000	0.8714±0.0000
	CDMH [33]	0.8322±0.0000	0.8798±0.0000	0.8214±0.0000	0.8230±0.0000	0.8214±0.0000
Hypergraph neural networks	HGNN [6]	0.8993±0.0056	0.9151±0.0080	0.8937±0.0054	0.8971±0.0056	0.8937±0.0054
	HGNN+ [7]	0.9114±0.0073	0.9239±0.0053	0.9065±0.0078	0.9097±0.0076	0.9065±0.0078
	HyperGCN [10]	0.8926±0.0337	0.8988±0.0327	0.8890±0.0392	0.8912±0.0391	0.8890±0.0392
	HNHN [34]	0.9127±0.0171	0.9237±0.0121	0.9082±0.0183	0.9111±0.0178	0.9082±0.0183
	EHGNN	**0.9664±0.0153**	**0.9681±0.0124**	**0.9651±0.0163**	**0.9662±0.0157**	**0.9651±0.0163**

a) The best results are marked in bold.

表 7 大规模数据集上不同方法的运行内存/显存 (GB)

Table 7 Running memory (GB) for different methods on large-scale datasets

Dataset	NN [28]	LPA [29]	DeepWalk [30]	EHGNN
Amazon-reviews	141(CPU)	231(CPU)	234(CPU)	46(CPU) + 12(GPU)
CIKM19-ECOMM	177(CPU)	352(CPU)	OOM	50(CPU) + 23(GPU)

方法的时间消耗情况. Node2Vec、传统超图学习方法、图/超图神经网络等方法如前所述, 会发生内存/显存溢出, 导致无法运行. NN 和 LPA 方法为直推学习方法, 不区分训练和测试阶段, 整体耗时也较少. 然而, 直推学习方法每次遇到新数据时均需要重新训练整个模型, 因而泛化性较差. DeepWalk 方法和 EHGNN 方法均为归纳学习方法, 训练阶段耗时在同一数量级, 而 DeepWalk 在测试阶段的耗时较短. EHGNN 在模型设计上相较 DeepWalk 更复杂, 因而耗时更长. 然而 EHGNN 的分类性能远高于 DeepWalk, 而在百万级别数据上测试阶段耗时也在半小时以内, 属于可接受的范围.

图 6 进一步展示了 EHGNN 与对比方法在 4 个数据集上的空间开销 – 准确率分布. 从图中可以发现, 在两个大规模数据集 Amazon-reviews 和 CIKM19-ECOMM 上, EHGNN 模型在分类精度和空间开销两方面均取得了优势 —— 分类精度最高, 内存占用最小. 而在两个小规模数据集 Walmart-trips

表 8 大规模数据集上不同方法的运行时间

Table 8 Running time for different methods on large-scale datasets

Dataset	Stage	NN [28]	LPA [29]	DeepWalk [30]	EHGNN
Amazon-reviews	Testing	7 h 57 min 26 s	1 d 1 h 25 min 2 s	13 s	25 min 34 s
	Training	–	–	1 d 21 h 5 min 38 s	4 d 1 h 40 min 31 s
CIKM19-ECOMM	Testing	13 h 40 min 51 s	17 h 13 min 30 s	OOM	22 min 49 s
	Training	–	–	OOM	3 d 1 h 41 min 27 s

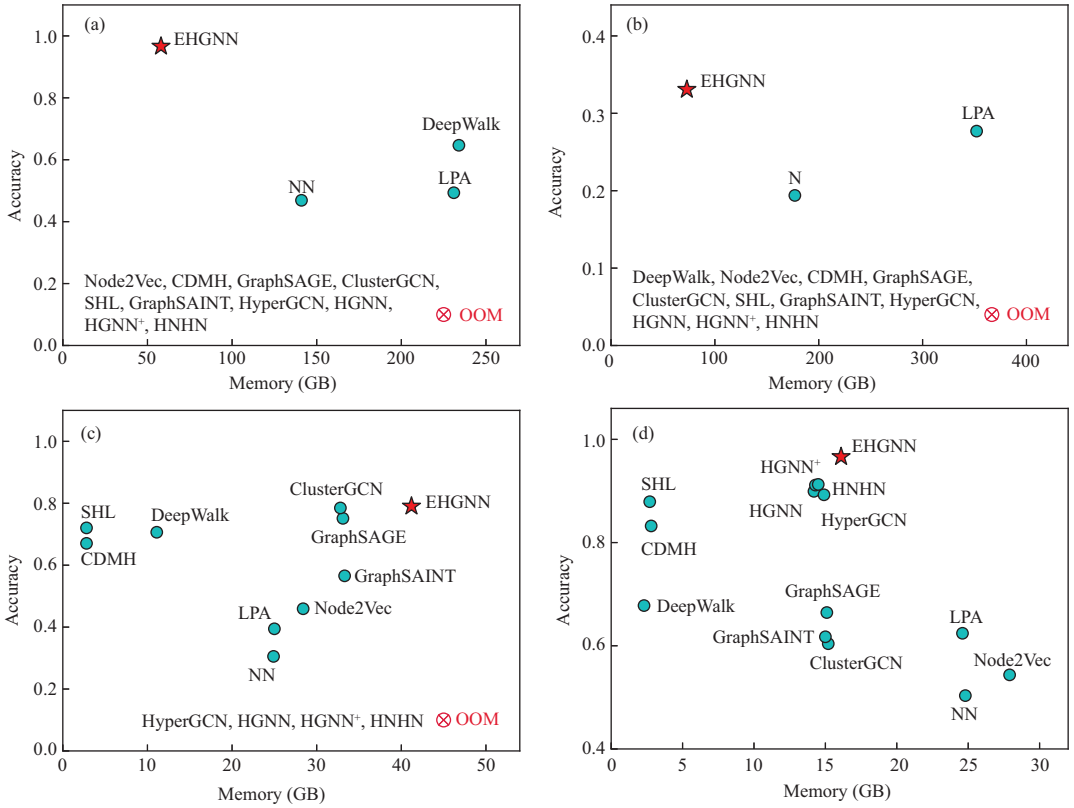

图 6 (网络版彩图) 4 个数据集上各方法的空间开销 – 准确率分布

Figure 6 (Color online) Space cost-accuracy distribution of different methods on four datasets. (a) Amazon-reviews; (b) CIKM19-ECOMM; (c) Walmart-trips; (d) House-bills

和 House-bills 上, EHGNN 仍在所有方法中取得了最好的预测性能. 在空间开销方面, 虽然 EHGNN 模型的内存占用要稍高于部分对比方法, 但所占用空间均属于同一数量级, 且为主流商用服务器能够支持的性能范围, 并不会制约方法应用. 以上实验结果表明, 在空间开销成为关键制约因素的大规模数据集上, EHGNN 在分类精度和空间开销两方面均能取得显著优势; 而在小规模数据集上, EHGNN 也能够拥有最优分类性能.

4.2 计算加速模块消融实验

为验证所提出的基于单阶段超图卷积的计算加速模块的有效性, 本小节在 Walmart-trips 数据集上进行了对应模块的消融实验, 对比方法包括所提出的 ENGNN 模型 (不包含计算加速模块)、包含

表 9　超图计算加速模块消融实验结果

Table 9　Ablation study on hypergraph computing acceleration module

Method	Running time for one training epoch (s)	Running time for one testing epoch (s)	Total running time	#Epochs
GraphSAGE	10.17	2.99	438 min 41 s	2000
EHGNN	27.90	3.93	53 min 4 s	100
EHGNN-fast	9.61	1.43	18 min 24 s	100

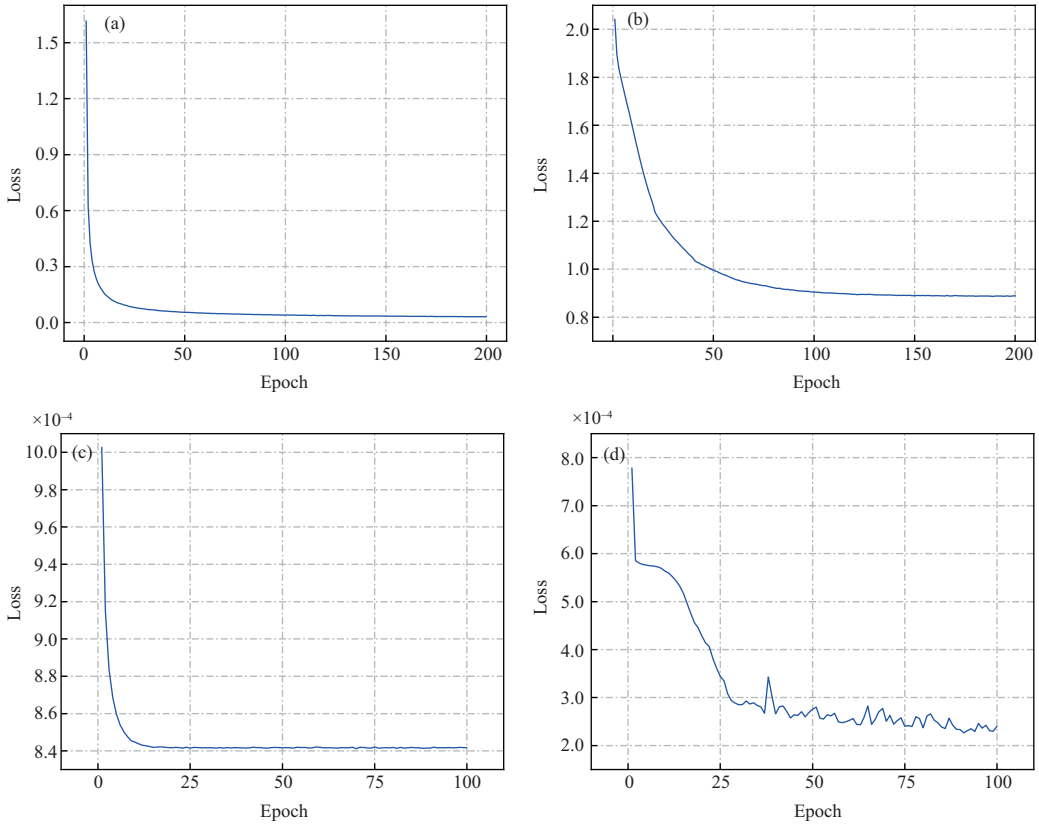

图 7　(网络版彩图) EHGNN 模型在 4 个数据集上的损失收敛曲线

Figure 7　(Color online) Convergence curves of EHGNN on four datasets. (a) Amazon-reviews; (b) CIKM19-ECOMM; (c) Walmart-trips; (d) House-bills

计算加速模块的 EHGNN 模型 EHGNN-fast, 以及最常见的图神经网络方法 GraphSAGE. 实验结果如表 9 所示.

从实验结果中, 有如下观察: (1) EHGNN-fast 能够在不影响收敛速度、不损失过多精度的前提下相较 EHGNN 加速约 3 倍; (2) 与 GraphSAGE 相比, EHGNN 在训练和测试阶段的单个 epoch 耗时均远高于 GraphSAGE, 约为 2 倍, 但由于 EHGNN 能较快达到收敛, 因此其计算总用时要显著低于 GraphSAGE, EHGNN-fast 进一步加速了模型训练和测试, 单个 epoch 耗时和总用时均优于 GraphSAGE. EHGNN-fast 的时间开销低于 GraphSAGE 的主要原因有二: 首先, GraphSAGE 基于超图数据的团扩展简单图来进行学习, 其边数量远远大于原超图中的超边数量, 导致 GraphSAGE 在进

行图卷积时要处理更多的节点关联; 其次, 团扩展得到的简单图结构相较原超图结构有信息损失, 因此 GraphSAGE 需要更多的训练轮次来达到收敛.

4.3 收敛性实验

为进一步说明所提出的 EHGNN 模型的收敛性能, 我们进行了收敛性实验. 图 7 展示了 EHGNN 模型在 4 个真实世界超图数据集上的收敛曲线. 从实验结果中, 可以观察到, EHGNN 在训练时能够较快收敛. 特别地, EHGNN 模型在 Amazon-reviews, CIKM19-ECOMM, Walmart-trips 和 House-bills 等数据集上分别在约第 50, 150, 10 和 50 个训练轮次即开始收敛.

5 总结与未来工作

本文针对现有超图神经网络面对大规模数据应用场景下时空复杂度过高的挑战, 提出了高效超图神经网络模型 EHGNN. EHGNN 针对现有方法空间开销大、计算效率低的问题, 分别设计了超图采样模块和基于单阶段超图卷积的计算加速模块. 其中, 超图采样模块又包含了分层采样算法和子超图预采样算法, 将难以进行整图计算的大规模超图分解为多个可处理的小规模子超图, 实现大规模数据的分批计算; 基于单阶段超图卷积的计算加速模块将传统的两阶段超图卷积优化为单阶段超图卷积, 避免了额外计算超边特征所带来的计算开销. 在 4 个真实超图数据集上的实验结果和对比验证了所提出 EHGNN 模型的高效性和有效性. 针对大规模数据设计超图神经网络仍有许多可探索的方向, 例如, 如何建模具有属性信息的大规模超图数据并设计相应的超图神经网络进行节点表示学习等.

参考文献

1 Ahn Y Y, Bagrow J P, Lehmann S. Link communities reveal multiscale complexity in networks. Nature, 2010, 466: 761–764

2 Fu G Y, Yu G X, Wang J, et al. Novel protein-function prediction using a directed hybrid graph. Sci Sin Inform, 2016, 46: 461–475 [傅广垣, 余国先, 王峻, 等. 基于有向混合图的蛋白质新功能预测. 中国科学: 信息科学, 2016, 46: 461–475]

3 Gao Y, Zhang Z Z, Lin H J, et al. Hypergraph learning: methods and practices. IEEE Trans Pattern Anal Mach Intell, 2022, 44: 2548–2566

4 Xia L H, Huang C, Xu Y, et al. Hypergraph contrastive collaborative filtering. In: Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022. 70–79

5 Nguyen D A, Nguyen C H, Petschner P, et al. SPARSE: a sparse hypergraph neural network for learning multiple types of latent combinations to accurately predict drug-drug interactions. Bioinformatics, 2022, 38: 333–341

6 Feng Y F, You H X, Zhang Z Z, et al. Hypergraph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2019. 3558–3565

7 Gao Y, Feng Y F, Ji S Y, et al. HGNN$^+$: general hypergraph neural networks. IEEE Trans Pattern Anal Mach Intell, 2023, 45: 3181–3199

8 Bai S, Zhang F H, Torr P H S. Hypergraph convolution and hypergraph attention. Pattern Recognit, 2021, 110: 107637

9 Veličković P, Cucurull G, Casanova A, et al. Graph attention networks. In: Proceedings of the International Conference on Learning Representations, 2018

10 Yadati N, Nimishakavi M, Yadav P, et al. HyperGCN: a new method of training graph convolutional networks on hypergraphs. In: Proceedings of the International Conference on Neural Information Processing Systems, 2019. 1511–1522

11 Ying R, He R N, Chen K F, et al. Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018.

974–983
12 Hamilton W, Ying R, Leskovec J. Inductive representation learning on large graphs. In: Proceedings of the International Conference on Neural Information Processing Systems, 2017. 1025–1035
13 Chiang W L, Liu X Q, Si S, et al. Cluster-GCN: an efficient algorithm for training deep and large graph convolutional networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019. 257–266
14 Zhou D Y, Huang J Y, Schölkopf B. Learning with hypergraphs: clustering, classification, and embedding. In: Proceedings of the International Conference on Neural Information Processing Systems, 2006. 1601–1608
15 Chen C, Li J, Zhou H Y, et al. Relation matters: foreground-aware graph-based relational reasoning for domain adaptive object detection. IEEE Trans Pattern Anal Mach Intell, 2023, 45: 3677–3694
16 Xi H, He L, Zhang Y, et al. Differentiable road pricing for environment-oriented electric vehicle and gasoline vehicle users in the bi-objective transportation network. Transp Lett, 2022, 14: 660–674
17 Jing Y C, Yang Y D, Wang X C, et al. Amalgamating knowledge from heterogeneous graph neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021. 15709–15718
18 Fan H Y, Zhang F B, Wei Y X, et al. Heterogeneous hypergraph variational autoencoder for link prediction. IEEE Trans Pattern Anal Mach Intell, 2022, 44: 4125–4138
19 Yu J L, Yin H Z, Li J D, et al. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In: Proceedings of the Web Conference, 2021. 413–424
20 Di D L, Li S R, Zhang J, et al. Ranking-based survival prediction on histopathological whole-slide images. In: Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention, 2020. 428–438
21 Wang J C, Zhang Y, Wei Y, et al. Metro passenger flow prediction via dynamic hypergraph convolution networks. IEEE Trans Intell Transp Syst, 2021, 22: 7891–7903
22 Jin T S, Cao L J, Zhang B C, et al. Hypergraph induced convolutional manifold networks. In: Proceedings of the International Joint Conference on Artificial Intelligence, 2019. 2670–2676
23 Ni J M, Li J C, McAuley J. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing, 2019. 188–197
24 Yang L W, Xiao Z B, Jiang W, et al. Dynamic heterogeneous graph embedding using hierarchical attentions. In: Proceedings of the European Conference on Information Retrieval, 2020. 425–432
25 Amburg I, Veldt N, Benson A. Clustering in graphs and hypergraphs with categorical edge labels. In: Proceedings of the Web Conference, 2020. 706–717
26 Chodrow P S, Veldt N, Benson A R. Generative hypergraph clustering: from blockmodels to modularity. Sci Adv, 2021, 7: eabh1303
27 Agarwal S, Branson K, Belongie S. Higher order learning with graphs. In: Proceedings of the International Conference on Machine Learning, 2006. 17–24
28 McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys, 1943, 5: 115–133
29 Raghavan U N, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E, 2007, 76: 036106
30 Perozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014. 701–710
31 Grover A, Leskovec J. Node2Vec: scalable feature learning for networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 855–864
32 Zeng H Q, Zhou H K, Srivastava A, et al. GraphSAINT: graph sampling based inductive learning method. In: Proceedings of the International Conference on Learning Representations, 2020
33 Zhang Z Z, Lin H J, Zhu J J, et al. Cross diffusion on multi-hypergraph for multi-modal 3D object recognition. In: Proceedings of the Pacific Rim Conference on Multimedia, 2018. 38–49
34 Dong Y H, Sawin W, Bengio, Y. HNHN: hypergraph networks with hyperedge neurons. In: Proceedings of the International Conference on Machine Learning Graph Representation Learning and Beyond Workshop, 2022. 1–11

Efficient hypergraph neural network on million-level data

Shuyi JI[1,3,4,5], Yuxuan WEI[1,3,4,5], Qionghai DAI[2,3,4,5] & Yue GAO[1,3,4,5*]

1. *School of Software, Tsinghua University, Beijing* 100084*, China*;
2. *Department of Automation, Tsinghua University, Beijing* 100084*, China*;
3. *Beijing Laboratory of Brain and Cognitive Intelligence, Beijing* 100084*, China*;
4. *Beijing National Research Center for Information Science and Technology, Beijing* 100084*, China*;
5. *Institute for Brain and Cognitive Science, Tsinghua University, Beijing* 100084*, China*

* Corresponding author. E-mail: gaoyue@tsinghua.edu.cn

Abstract High-order correlations are ubiquitous in the real world, such as the social network, the biological network, and the transportation network. It is of significant importance to model and optimize high-order correlations for network investigation. The hypergraph, as a flexible and scalable structure, can be applied to model the high-order correlations in a natural manner. With the development of deep learning, hypergraph neural networks (HGNNs) are widely leveraged for high-order correlation modeling and optimization. Although existing HGNNs have shown decent performance on small-scale datasets, they cannot be applied to large-scale data due to their expensive space cost caused by the transductive learning paradigm in that case. This paper first analyzes the root causes of the deficiency that HGNNs are unable to handle large-scale data. Furthermore, this paper presents the efficient hypergraph neural network (EHGNN) towards the million-level data. EHGNN designs the hypergraph sampling module and the computational acceleration module that is based on single-stage hypergraph convolution, reducing the time and space cost of HGNNs. Experimental results on four real-world hypergraph datasets demonstrate the effectiveness and efficiency of the proposed EHGNN.

Keywords hypergraph computation, hypergraph neural network, high-order correlation, large-scale data, vertex classification