基于超图神经网络的链路预测方法

陈亮[1]，赵英[1]，史晟辉[1]，尹琳[2]
（1.北京化工大学信息科学与技术学院，北京 100029；2.中日友好医院远程医疗中心，北京 100029）

摘　要：随着信息技术的飞速发展，链路预测技术已经在多个领域得到了广泛的应用。目前的链路预测方法通常采用子图提取的方式，其中一种基于线图转换（LGT）与图卷积神经网络（GCN）的模型在链路预测问题上取得了优异的效果，但仍存在 2 个问题：1）LGT 的时间复杂度过高和转换后子图的规模过大导致其难以被广泛应用；2）GCN 忽略了节点间的高阶关系和局部聚类结构，会对预测精度产生一定的影响。为解决上述问题，提出一种基于超图卷积神经网络（HGCN）的链路预测方法 HGLP。该方法使用对偶超图转换（DHT）替代 LGT 以做到在不损失任何结构信息的情况下提高系统的运行效率，同时运用 HGCN 分别学习超图中超节点与超边的高阶特征以实现更高的预测精度。实验结果表明，在曲线下面积（AUC）和平均准确率（AP）2 个指标下，所提出的方法在 7 种不同领域的真实图数据集中的表现不仅优于现有的链路预测方法，而且内存占用更少、运行时间更短。

关键词：链路预测；超图；超图神经网络；对偶超图转换；深度学习
源代码链接：https://github.com/634407371/HGLP
中图分类号：TP181　　**文献标志码：**A　　**DOI：**10.19678/j.issn.1000-3428.0069952

Link Prediction Method Based on Hypergraph Neural Network

CHEN Liang[1], ZHAO Ying[1], SHI Shenghui[1], YIN Ling[2]
(1.College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China;
2.Telemedicine Center, China-Japan Friendship Hospital, Beijing 100029, China)

【Abstract】With the rapid development of information technology, link prediction has been widely applied in various fields. Current link prediction methods are based on subgraph extraction. Models based on Line Graph Transformation (LGT) and Graph Convolutional Network (GCN) achieve excellent results in link prediction. However, two problems remain: 1) the high time complexity of the LGT and the large size of the line graph hinder its wide-spread application; 2) GCN ignores the high-order relationship and local clustering structure between nodes, thereby affecting prediction accuracy. To solve the above issues, this paper proposes a link prediction method based on Hypergraph Convolutional Network (HGCN), called HGLP. This method replaces LGT with Dual Hypergraph Transformation (DHT) to improve system efficiency without losing structural information and applies HGCN to learn the higher-order features of the hypernodes and hyperedges in the hypergraph to obtain higher prediction accuracy. Experimental results show that the proposed method outperforms state-of-the-art link prediction methods on seven real-world datasets from different domains, in terms of Area Under the Curve (AUC) and Average Precision (AP). Furthermore, the proposed method achieves shorter runtimes and less memory usage.

【Key words】link prediction; hypergraph; Hypergraph Convolutional Network (HGCN); Dual Hypergraph Transformation (DHT); deep learning

0　引言

链路预测是根据已知的链路信息预测 2 个实体之间存在链路的可能性大小。这里的链路可以因实体的变化而抽象为各种各样的关系，如人与人之间的好友关系、分子与分子间的反应关系等。链路的这种特性使得链路预测技术在实际生活中具有广泛的应用场景[1]，如：社交软件的好友推荐、购物平台的商品推荐、搜索页面的词条推荐、新闻资讯的热点推荐、个性化的广告推送以及知识图谱补全等[2-3]，其在方便人们生活的同时也蕴含着巨大的商业价值。如何进一步提升链路预测的准确性和计算效率，已经成为目前研究的热点问题。

链路预测往往离不开图。将实体定义为图中的

基金项目：中央高水平医院临床业务费专项成果转化项目（2023-NHLHCRF-YXHZ-MS-04）；北京化工大学-中日友好医院生物医学转化工程研究中心联合项目（XK2023-18）。
作者简介：陈亮，男，硕士研究生，主研方向为人工智能、链路预测；赵英（通信作者）、史晟辉，教授、博士；尹琳，副研究员、硕士。
收稿日期：2024-06-03　　**修回日期：**2024-07-18　**E-mail：**zhaoy@mail.buct.edu.cn

节点，实体之间的关系定义为图中的边，抽象的实体间关系预测问题即可转变为图的连边预测问题。但图是一种非欧氏数据结构[4]，而传统神经网络通常只能处理常规的欧氏结构数据[5-6]。为此，研究人员提出了一种新型神经网络——图神经网络(GNN)[7]，它能很好地解决这种非欧氏结构数据的处理问题。

近年来，GNN 在链路预测中得到了越来越广泛的应用[8-9]。基于子图提取的 GNN 方法是目前的一种主流链路预测方法，其中，LGLP[10]通过线图转换(LGT)取得了优异的链路预测性能，但仍存在着一些问题：LGT 的时间和空间复杂度较高，导致算法的计算量和内存占用过大，运行效率低，且其所使用的图卷积神经网络(GCN)[11]在信息传播时只考虑了节点是否相邻，而忽略了节点间的高阶关系和局部聚类结构[12]，对链路预测的精度有一定的影响。

为了解决上述问题，本文提出了一种新的链路预测方法 HGLP，主要工作如下：

1）使用时间复杂度更低的对偶超图转换(DHT)[13]替代 LGT，在保证原图结构信息不丢失的情况下缩短转换时间，且使得变换后的子图规模也大幅缩小，有效提升了系统的运行效率。

2）设计了一个基于超图卷积神经网络(HGCN)[12]的链路预测方法，通过 HGCN 分别学习节点和边的高阶特征，再将待预测链路的特征与链路 2 个端点的特征进行拼接得到用于二分类的子图特征，二分类结果即为链路预测结果，有效提升了系统的预测精度。

3）在 7 个常用的公共数据集上进行了实验。实验结果表明，HGLP 在曲线下面积(AUC)和平均准确率(AP)2 个指标下性能均优于所对比的基线方法，且运行时间更短、内存占用更少，证明了 HGLP 的有效性。

1 相关工作

链路预测通常可以理解为对图中不存在的边进行预测打分的过程[14]。一般得分越高，该链路存在的概率越大，常见的打分方法有启发式方法和基于神经网络的方法[15]。

1.1 启发式方法

启发式方法根据现实生活中的规律和相关指标而总结出一套公式或规则。如 CN[16]指标认为两节点共同邻居越多，其连接的概率越大。除此之外，还有 Jaccard[17]、PA[18]、RA[19]、AA[20]、Katz[21]、SimRank[22]、rooted PageRank[23]等。这些方法通常分为一阶启发式、二阶启发式和高阶启发式，它们的主要区别在于利用了多少阶邻域的信息进行分数的计算。一般情况下，利用的信息邻近阶数越高，预测精度越高，但也伴随着时间复杂度的升高。同时启发式方法还有一个共同的缺点——普适性较差[24]，即一个指标很难同时在不同数据集中保持相同的精度，如共同邻居 CN 指标在好友推荐中表现很好，但在分子间化学反应的预测中表现却不如人意[25]。

1.2 基于神经网络的方法

为了解决启发式方法普适性较差的问题，研究人员开始使用神经网络来学习一个自适应的链路预测方法。WLNM[24]通过提取待预测链路 h 跳邻域子图，使用 PALETTE-WL 着色算法对子图顶点进行排序，将根据排序得到的邻接矩阵输入多层感知机(MLP)中进行链路预测。由于 MLP 限制了输入邻接矩阵的大小，因此需要删除一些排序靠后的节点，这导致子图并非完整的 h 跳邻域，造成了部分信息的缺失[25]。为了提取完整子图，SEAL[25]采用深度图卷积神经网络(DGCNN)[26]，并根据与待预测链路 2 个端点的距离来学习节点特征，随后再对学习后的特征进行池化操作，挑选固定数量的特征来表示整个子图，最后对子图进行二分类。尽管其预测效果优于 WLNM，但由于采用的池化操作导致了原始子图的部分信息丢失，影响了预测的精度[10]。为了规避池化操作带来的问题，LGLP 运用 LGT 方法将边转换为节点，将节点转换为边，然后利用 GCN 学习待预测链路对应节点的特征，并将该特征输入到 MLP 中进行链路预测。规避池化操作后的 LGLP 性能虽然得到了进一步提升，但仍然存在着时间复杂度与空间复杂度过高的问题，并且 SEAL 和 LGLP 中使用的 GCN 忽略了节点之间的高阶关系和局部聚类结构，给预测精度带来了较大的影响。

2 方法

2.1 问题定义

给定一个无向无权图 $G=(V,E,\boldsymbol{A})$，其中有 n 个节点、m 条边。V、E 分别为图中所有节点和边的集合，$|V|=n$，$|E|=m$。$\boldsymbol{A}\in\{0,1\}^{n\times n}$ 为图的邻接矩阵。链路预测的目标就是根据观察到的 V 和 E，推断出一对目标节点 u，$v(u,v\in V)$之间是否存在一条边[27]。

2.2 框架

HGLP 的整体框架如图 1 所示，其中线条加粗

部分代表待预测链路及其 2 个端点。对于每一条待预测的连边，它首先提取以其 2 个端点为中心的完整 h 跳邻域子图，然后对子图中每个节点使用节点标注算法进行标注，将标注转为独热编码的形式以得到节点特征，再对边的两端点特征进行拼接来得到边特征。之后采用 DHT 将标注后的子图的边转换成超节点，将节点转换成超边，此时超边的特征即为原始子图节点特征，超节点特征即为原始子图边的特征。通过转置超图关联矩阵，使用 2 个相互独立的 HGCN 分别学习超节点与超边的特征，再提取待预测链路及其 2 个端点的特征进行拼接，最后输入 MLP 模块进行二分类，以得到预测结果。整体框架由以下 5 个模块组成：子图提取模块、特征生成模块、对偶超图转换模块、特征学习模块和链路预测模块。

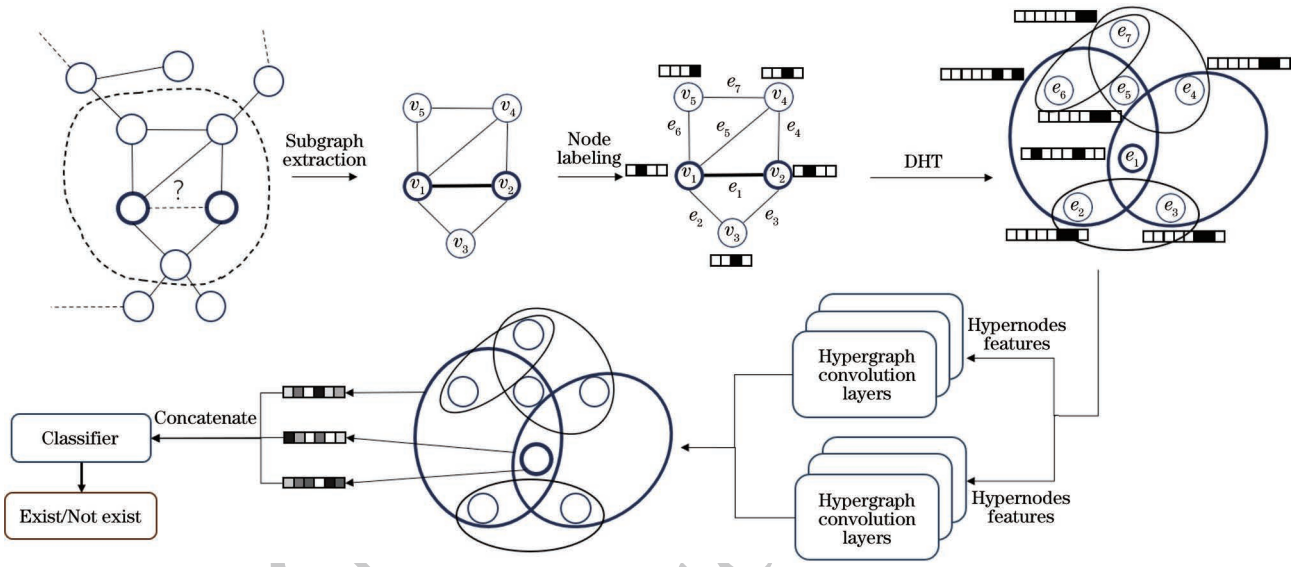

图 1 HGLP 模型框架

Fig. 1 The framework of HGLP model

2.2.1 子图提取模块

在子图提取模块中，需要分别对每一条待预测的边提取一个完整的 h 跳邻域子图，该子图包含了待预测连边的 2 个端点 u，v(u，$v\in V$)与 2 个端点的所有 h 跳邻居。h 跳邻居的定义如下：

$$N^h=\{x\mid \min(d(x,u),d(x,v))\leqslant h\} \tag{1}$$

式中：$d(x,y)$代表节点 x 和 y 之间的最短距离；N 代表邻居节点集合；h 代表第 h 跳。

设子图为 $G'=(V',E',\boldsymbol{A}')$，则：

$$V'=N^h \tag{2}$$

$$E'=\{(x,y)\mid (x,y)\in E,x\in V',y\in V'\} \tag{3}$$

式中：V'代表子图的节点集合；E'代表子图的边集合；$\boldsymbol{A}'$代表子图的邻接矩阵。

图 1 中所示即为提取待预测链路 2 个端点(加粗的 2 个节点)的 1 跳邻域子图。

2.2.2 特征生成模块

特征生成模块需要对子图中的每个节点生成一个用于神经网络学习的特征向量，该特征需要能够让神经网络准确地识别出目标节点(待预测连边的 2 个端点)以及其他节点在网络中的结构重要性。本文采用的方法为 SEAL 中的双半径节点标注法。该方法首先在不考虑待预测连边的情况下分别计算每个节点距离 2 个目标节点的最短距离，然后使用如下所示的哈希函数为每个节点进行标注：

$$L=1+\min(d_u,d_v)+\left(\frac{d}{2}\right)\left[\left(\frac{d}{2}\right)+(d\%2)-1\right] \tag{4}$$

式中：$d=d_u+d_v$；$d_u=d(x,u)$；$x\in V'$；u 和 v 为目标节点；/代表整除；%代表取余；L 代表标注结果，为一个大于 1 的整数。此外，对于 $d=\infty$的节点，设定 $L=0$；对于 2 个目标节点，设定 $L=1$，此操作能够让神经网络在训练过程中有效地识别出目标节点。最后将整数转换为独热编码向量，即可得到节点特征。

以图 1 中的节点 v_5 为例，v_1、v_2 为目标节点，v_5 至 v_1 的最短路径为 $v_5\rightarrow v_1$，最短距离为 1；v_5 至 v_2 的最短路径为 $v_5\rightarrow v_4\rightarrow v_2$，最短距离为 2，其标注为 $1+\min(1,2)+\left(\frac{3}{2}\right)\left[\left(\frac{3}{2}\right)+(3\%2)-1\right]=3$，转为独热编码形式后特征为“0001”。

2.2.3 对偶超图转换模块

针对现有方法中 LGT 过程时间复杂度过高、转换后图规模过大的问题，本文采用 DHT 进行替

代。超图是图的泛化，一条边可以连接任意数量的节点，能很好地表示节点之间的高阶关系[28]。

超图的定义如下：

$$G_{HG}=(V,E,\boldsymbol{H}) \tag{5}$$

式中：V、E 分别代表所有超节点和超边的集合；$\boldsymbol{H}$ 代表超图关联矩阵。设超图中的节点数和边数分别为 n 和 m，则 $|V|=n$，$|E|=m$，$\boldsymbol{H}\in\{0,1\}^{n\times m}$[29]。

与 LGT 类似，DHT 会将原图中的节点转换为超图中的超边，原图中的边转换为超图中的超节点。由于超图是图的泛化，因此原始图也可以表示为一种每条超边固定连接 2 个超节点的超图：

$$G=(V,E,\boldsymbol{A})\rightarrow G_{HG}=(V,E,\boldsymbol{H}) \tag{6}$$

DHT 可表示为：

$$G_{HG}=(V,E,\boldsymbol{H})\rightarrow G'_{HG}=(E,V,\boldsymbol{H}^{T}) \tag{7}$$

此转换过程从宏观上看仅仅是对超图关联矩阵进行了转置操作，因此不会导致原图中任何结构信息的丢失，且其实际算法的时间复杂度仅为 $O(m)$，相较于 LGT 的时间复杂度 $O(m^2)$，其转换时间将大幅缩短。具体转化过程示例如图 2 所示。假设原图有 5 个节点、6 条边，经过 LGT 操作后，节点数变为 6，边数变为 10，而经过 DHT 操作后，节点数为 6，边数为 5。相比于 LGT，DHT 为双射变换，具有不变性，即 2 个不同的图转换后仍然是不同的，且不会丢失原图中的任何结构信息，易于实现，同时时间和空间复杂度更低[13]。

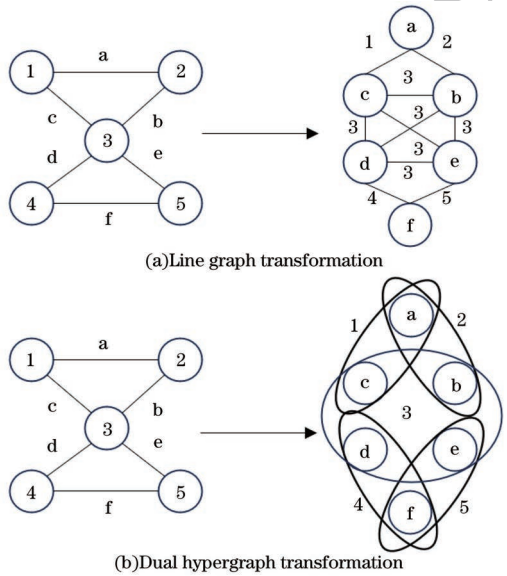

图 2 线图转换与对偶超图转换

Fig. 2 Line graph transformation and dual hypergraph transformation

2.2.4 特征学习模块

由于超图关联矩阵结构的特殊性，可以很容易地通过转置操作将超节点与超边进行互换[30]，因此在特征学习模块中，可以通过转置操作分别学习超节点与超边的高阶特征。在 2.2.2 节中，已经获得了原图中的节点特征，原图中的边特征可以通过式(8)得到：

$$\boldsymbol{X}_{(i,j)}=\min(\boldsymbol{X}_i,\boldsymbol{X}_j)\,\|\,\max(\boldsymbol{X}_i,\boldsymbol{X}_j) \tag{8}$$

式中：$(i,j)\in E$，$i,j\in V$；min 和 max 分别代表获取特征向量对应位置的最小值和最大值操作；“|”表示拼接操作；$\boldsymbol{X}_i$ 表示节点 i 的特征；$\boldsymbol{X}_{(i,j)}$ 表示边 (i,j) 的特征。

以图 1 中的边 e_5 为例，其在原始子图中的 2 个端点分别为 v_1 与 v_4，其特征分别为“0100”“0010”，因此根据式(8)拼接后，e_5 的特征为“00000110”

经过 DHT 后，超节点和超边的特征分别对应原图的边和节点的特征。然后通过 2 个相互独立的 HGCN 分别学习超节点和超边的特征。

HGCN 的定义如下所示[12]：

$$\boldsymbol{X}^{l+1}=\sigma(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}\boldsymbol{W}\boldsymbol{B}^{-1}\boldsymbol{H}^{T}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{X}^{l}\boldsymbol{P}) \tag{9}$$

式中：$\boldsymbol{X}$ 为特征矩阵；$\boldsymbol{P}$ 为可学习的权重矩阵；$\boldsymbol{W}\in\mathbb{R}^{m\times m}$ 为边的权重矩阵(若无边权重，则其为单位矩阵)；$\boldsymbol{H}$ 为超图关联矩阵；σ 为 ReLU 激活函数。

$$f_{ReLU}(x)=\max(0,x) \tag{10}$$

$\boldsymbol{D}\in\mathbb{R}^{n\times n}$ 为节点度矩阵，$\boldsymbol{B}\in\mathbb{R}^{m\times m}$ 为边度矩阵。$\boldsymbol{D}$ 和 $\boldsymbol{B}$ 的定义如下所示：

$$\boldsymbol{D}_{ii}=\sum_{\varepsilon=1}^{m}\boldsymbol{W}_{\varepsilon\varepsilon}\boldsymbol{H}_{i\varepsilon} \tag{11}$$

$$\boldsymbol{B}_{ii}=\sum_{i=1}^{n}\boldsymbol{H}_{i\varepsilon} \tag{12}$$

此外，为了放大每个节点或边自身的特征，本文对式(9)增加了一次自环操作以改进 HGCN[31]。设超节点特征为 $\boldsymbol{X}_v$，超边特征为 $\boldsymbol{X}_e$，经过 DHT 后的超图关联矩阵为 $\boldsymbol{H}$，则当学习超节点的特征时，改进的超图卷积层应为：

$$\boldsymbol{X}_v^{l+1}=\sigma((\boldsymbol{D}_v^{-\frac{1}{2}}\boldsymbol{H}\boldsymbol{W}_v\boldsymbol{B}_v^{-1}\boldsymbol{H}^{T}\boldsymbol{D}_v^{-\frac{1}{2}}+\boldsymbol{I})\boldsymbol{X}_v^{l}\boldsymbol{P}_v) \tag{13}$$

式中：$\boldsymbol{W}_v$ 为单位矩阵；$\boldsymbol{P}_v$ 为用于学习超节点特征的权重矩阵；$\boldsymbol{D}_v$ 和 $\boldsymbol{B}_v$ 由 $\boldsymbol{W}_v$ 和 $\boldsymbol{H}$ 根据式(11)、式(12)计算得到。

当学习超边特征时，改进的超图卷积层应为：

$$\boldsymbol{X}_e^{l+1}=\sigma((\boldsymbol{D}_e^{-\frac{1}{2}}\boldsymbol{H}^{T}\boldsymbol{W}_e\boldsymbol{B}_e^{-1}\boldsymbol{H}\boldsymbol{D}_e^{-\frac{1}{2}}+I)\boldsymbol{X}_e^{l}\boldsymbol{P}_e) \tag{14}$$

式中：$\boldsymbol{W}_e$ 为单位矩阵；$\boldsymbol{P}_e$ 为用于学习超边特征的权重矩阵；$\boldsymbol{D}_e$ 和 $\boldsymbol{B}_e$ 由 $\boldsymbol{W}_e$ 和 $\boldsymbol{H}^{T}$ 根据式(11)、式(12)计算得到。整个实现过程如图 3 所示，由每层的输出经过拼接操作得到最终的输出特征。

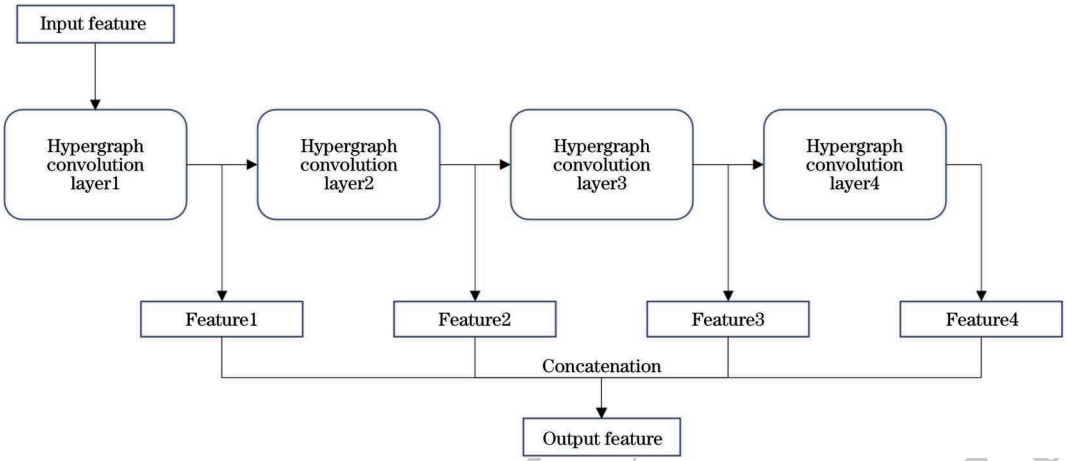

图 3 HGLP 的超图卷积网络结构

Fig. 3 The structure of hypergraph convolution network in HGLP

2.2.5 链路预测模块

对于每条要预测的连边，提取其对应的超节点特征以及其 2 个端点对应的超边特征，由式(15)得到表示子图的特征：

$$\boldsymbol{X}_{G'}=\min(\boldsymbol{X}_i,\boldsymbol{X}_j)\,\|\,\max(\boldsymbol{X}_i,\boldsymbol{X}_j)\,\|\,\boldsymbol{X}_{(i,j)} \quad (15)$$

最终将 $\boldsymbol{X}_{G'}$ 输入 MLP 中进行二分类，分类结果将对应有边和无边。

3 实验

为了验证 HGLP 在链路预测中的表现，本文采用曲线下面积(AUC)和平均准确率(AP)作为评测基准指标，并在业界常用的 7 种不同领域的数据集中进行实验[10]。实验平台 CPU 为 i7-9750H，GPU 为 GTX 1660Ti MQ，内存为 16 GB，显存为 6 GB，操作系统为 Windows 10，运行环境为 Python 3.10.13、PyTorch 2.0.0 和 PyTorch-geometric 2.5.1。

3.1 数据集

本文使用的所有数据集如表 1 所示。所有数据集均为真实的图数据集，且来自不同领域，能够很好地验证 HGLP 的普适性。其中，Celegans[32] 为线虫的神经网络，USAir[33] 为美国航空网络，SMG 和 GRQ 为合著网络，EML 为邮件共享网络，YST[34] 为蛋白质相互作用关系的网络，Power[32] 为美国西部电力网络。对于每个数据集，提取 80%数量的边以及等量不存在的边(随机抽取)作为训练集，剩余 20%的边与同样等量的不存在的边(随机抽取)作为测试集。每个数据集以不同的随机数种子运行 10 遍取平均值作为最终结果。

表 1 数据集

Table 1 Datasets

数据集	节点数/个	边数/条	平均节点度
Celegans	297	2 148	14.46
USAir	332	2 126	12.81
SMG	1 024	4 916	9.60
EML	1 133	5 451	9.62
YST	2 284	6 646	5.81
Power	4 941	6 594	2.67
GRQ	5 241	14 484	5.53

3.2 基线方法

本文选取以下 9 个基线方法与 HGLP 进行比较：一阶启发式方法 CN；二阶启发式方法 AA；高阶启发式方法 Katz、SimRank(SR)和 rooted PageRank(PR)；传统图神经网络方法 GCN 和 GAT；基于子图提取的图神经网络的方法 SEAL 和 LGLP。

3.3 参数设定

Katz 中的衰减系数为 0.001。SimRank 中的重要性系数为 0.8。rooted PageRank 的衰减系数为 0.85。SEAL 固定提取 2 阶邻域子图，4 个图卷积层通道大小分别设置为 32、32、32 和 1，池化层的比例设为 0.6，2 个一维卷积层通道大小分别设置为 16 和 32，2 个全连接层通道大小分别设置为 128 和 2，学习率为 0.000 1，批次大小为 50，最大训练次数为 50。LGLP 也固定提取 2 阶邻域子图，4 个图卷积层通道大小分别设置为 32、32、32 和 1，2 个全连接层通道大小分别设置为 128 和 2，学习率为 0.005，批次大小为 50，最大训练次数为 15。GCN 和 GAT 均为 4 个图神经网络层与 2 个全连接层的组合结构，通道大小与 LGLP 保持一致，学习率为 0.005，最大训练次数为 200，其中 GAT 为单头注意力。

对于本文提出的 HGLP，固定提取 2 阶邻域子图，用于学习超边特征的 4 个图卷积层通道大小分别设置为 32、32、32 和 1，用于学习超节点特征的

4 个图卷积层通道大小分别设置为 64、64、64 和 2，2 个全连接层通道大小分别设置为 128 和 2，学习率为 0.005，批次大小为 50，最大训练次数为 15，损失函数为 NLLLoss。

3.4 链路预测性能与效率验证

为验证 HGLP 的链路预测性能，本文进行了对比实验。AUC 指标下的平均值和标准差如表 2 所示，AP 指标下的平均值和标准差如表 3 所示（其中加粗数据表示最优值，下同）。实验结果表明：启发式方法普适性较差，如 CN 在 USAir 数据集中 2 种指标能够达到 90%以上，而在 Power 数据集中却只有 60%左右，对于不同数据集的表现差异非常大；基于图神经网络的 4 种方法以及本文提出的 HGLP 通过使用神经网络来学习图的特征，不仅具有很好的普适性，而且实验结果绝大部分优于启发式方法；对比传统图神经网络方法，基于子图提取的 3 种方法明显更优，体现了子图提取应用于链路预测的有效性；而本文所提出的 HGLP 在两项指标下，在所有数据集上的表现均要优于基线方法 LGLP，实现了链路预测精度的进一步提升。

表 2 AUC 对比

Table 2 AUC comparison %

Model	Celegans	USAir	SMG	EML	YST	Power	GRQ
CN	82.94(±1.11)	92.82(±0.96)	81.15(±0.72)	82.04(±0.43)	68.47(±0.54)	57.29(±0.24)	89.60(±0.28)
AA	84.61(±1.04)	93.85(±0.98)	82.06(±0.71)	82.21(±0.43)	68.51(±0.55)	57.29(±0.24)	89.62(±0.27)
Katz	85.27(±1.22)	91.58(±1.19)	86.23(±0.62)	88.59(±0.61)	80.20(±0.36)	71.67(±1.15)	89.55(±0.43)
SR	76.17(±2.19)	79.59(±1.08)	78.00(±0.75)	87.14(±0.71)	74.05(±0.38)	60.72(±2.68)	89.52(±0.43)
Rooted PR	89.13(±1.09)	93.31(±1.26)	89.63(±0.67)	89.70(±0.58)	81.14(±0.36)	60.74(±2.59)	89.72(±0.42)
GCN	84.37(±1.18)	94.37(±0.87)	85.27(±0.79)	80.81(±0.67)	79.68(±0.53)	76.53(±1.64)	86.72(±0.88)
GAT	84.45(±0.90)	94.04(±0.94)	84.09(±1.06)	80.43(±1.14)	76.36(±0.73)	77.37(±1.02)	85.43(±0.81)
SEAL	88.70(±1.23)	96.40(±0.83)	91.69(±0.73)	91.18(±0.59)	89.94(±0.85)	81.64(±1.66)	96.95(±0.30)
LGLP	90.19(±0.84)	97.19(±0.56)	92.86(±0.57)	91.89(±0.49)	91.71(±0.36)	83.03(±0.70)	97.34(±0.14)
HGLP	**90.71**(±0.38)	**97.66**(±0.34)	**93.18**(±0.73)	**92.29**(±0.72)	**92.50**(±0.43)	**83.61**(±0.89)	**97.69**(±0.23)

表 3 AP 对比

Table 3 AP comparison %

Model	Celegans	USAir	SMG	EML	YST	Power	GRQ
CN	80.19(±1.13)	92.62(±0.87)	80.19(±0.70)	81.36(±0.53)	68.33(±0.54)	57.27(±0.25)	89.56(±0.28)
AA	83.95(±1.16)	94.40(±0.84)	83.07(±0.64)	82.30(±0.47)	68.73(±0.58)	57.27(±0.24)	89.65(±0.27)
Katz	84.77(±1.15)	93.47(±0.80)	88.29(±0.65)	90.60(±0.71)	85.66(±0.34)	73.71(±0.93)	93.00(±0.29)
SR	66.61(±2.35)	70.71(±1.39)	70.81(±1.57)	87.43(±0.93)	77.99(±0.66)	74.09(±1.04)	92.89(±0.24)
Rooted PR	88.02(±1.44)	94.08(±1.11)	91.33(±0.63)	91.24(±0.66)	86.25(±0.22)	74.63(±1.00)	93.13(±0.25)
GCN	82.50(±1.48)	94.40(±1.18)	84.54(±0.81)	79.71(±1.04)	80.06(±0.69)	71.97(±1.88)	86.38(±1.40)
GAT	81.95(±1.33)	93.63(±0.94)	82.42(±1.23)	78.40(±1.60)	75.94(±0.99)	72.70(±0.94)	84.02(±1.23)
SEAL	87.93(±1.21)	96.46(±0.87)	92.16(±0.66)	91.77(±0.79)	91.33(±0.75)	84.10(±1.32)	97.63(±0.24)
LGLP	89.57(±0.96)	97.25(±0.62)	93.27(±0.50)	92.63(±0.52)	92.77(±0.26)	85.31(±0.66)	97.95(±0.09)
HGLP	**90.13**(±0.57)	**97.73**(±0.40)	**93.64**(±0.59)	**92.86**(±0.63)	**93.29**(±0.42)	**85.57**(±0.75)	**98.17**(±0.17)

为了验证 HGLP 所采用的 DHT 以及 HGCN 在预测效率上对比 LGLP 所采用的 LGT 和 GCN 是否有提升，本文还进行了运行时间和内存占用的对比实验，对比结果如表 4 所示。其中：T1 代表转换过程的时间；LGLP 对应 LGT；HGLP 对应 DHT；T2 代表 GCN 单次训练的时间；RAM1 代表转换过程内存占用；RAM2 代表训练过程内存占用；VRAM 代表训练过程显存占用。上述内存均代表操作系统中提交的虚拟内存（物理内存和分页文件大小）总量，显存代表 GPU 专用显存和共享显存的总和。

对于所有的数据集，DHT 相比于 LGT 运行时间均得到有效缩短，其中在 SMG 数据集上的效果最好，缩短了 99.24%；而 Power 数据集由于其平均节点度过低、图过于稀疏使得效果最差，但转换时间也缩短了 92.20%。在内存占用方面，同样在除 Power 数据集以外的数据集中，DHT 所占用的内存都要明显少于 LGT，其中在 USAir 数据集上的效果最好，内存占用减少了 93.43%，可见使用时间复杂度更低的 DHT 代替 LGT 不仅能显著加速转换过程，而且在稠密图上其占用的内存也能显著降低。而转换过程的内存占用大小将直接反映转换后子图的规模大小，并且影响到 GCN 部分的训练时间以

及内存与显存的占用。在训练过程中，HGLP 虽然使用了更为复杂的 2 个 4 层的 HGCN 结构，但其单次训练时间依然要小于 LGLP，其中 SMG 数据集的单次训练时间下降最明显，达到了 81.77%。在训练过程的内存与显存占用方面，除 Power 数据集因其过于稀疏而无明显变化外，其余数据集均得到了内存与显存的下降，因此本文提出的 HGLP 模型能够在不损失预测精度的情况下，实现链路预测效率的大幅提升、内存占用的大幅下降，且更适用于稠密图的链路预测。

表 4 链路预测效率对比

Table 4 Link prediction efficiency comparison

Dataset	Model	T1/s	T2/s	RAM1/MB	RAM2/MB	VRAM/MB
Celegans	LGLP	65.780	8.73	2 985	5 246	5 275
	HGLP	0.760	4.43	335	815	417
USAir	LGLP	87.910	19.57	5 845	9 765	9 334
	HGLP	0.890	4.93	384	1 077	643
SMG	LGLP	300.224	67.87	13 011	4 921	6 563
	HGLP	2.280	12.37	976	1 460	1 011
EML	LGLP	130.756	12.33	2 590	3 082	2 665
	HGLP	1.780	9.89	483	872	441
YST	LGLP	115.769	12.22	1 576	1 670	1 262
	HGLP	1.730	11.02	285	699	246
Power	LGLP	18.080	6.46	19	439	61
	HGLP	1.410	5.64	23	475	57
GRQ	LGLP	163.120	29.69	5 788	2 784	2 333
	HGLP	3.580	23.09	536	916	275

3.5 消融实验

为了验证 HGLP 分别学习超节点与超边的方法是否有效，本文将只学习超节点特征的模型设为 HGLP-n，只学习超边特征的模型设为 HGLP-e，进行消融实验，实验结果如表 5 和表 6 所示。可以看到，HGLP 在所有数据集中 AUC 和 AP 均要高于 HGLP-n 和 HGLP-e，说明两者的结合能够有效地提升预测精度，同时 HGLP-e 的大部分结果均高于 HGLP-n，说明超边特征的学习对总体链路预测精度的贡献更大。

表 5 消融实验结果(AUC)

Table 5 Ablation experiment results (AUC) %

Model	Celegans	USAir	SMG	EML	YST	Power	GRQ
HGLP-n	89.97(±0.44)	97.30(±0.40)	92.67(±0.77)	92.01(±0.67)	92.21(±0.33)	83.28(±0.89)	97.61(±0.24)
HGLP-e	90.53(±0.43)	97.64(±0.38)	92.99(±0.74)	92.09(±0.62)	92.35(±0.46)	83.30(±0.90)	97.58(±0.23)
HGLP	**90.71**(±0.38)	**97.66**(±0.34)	**93.18**(±0.73)	**92.29**(±0.72)	**92.50**(±0.43)	**83.61**(±0.89)	**97.69**(±0.23)

表 6 消融实验结果(AP)

Table 6 Ablation experiment results (AP) %

Model	Celegans	USAir	SMG	EML	YST	Power	GRQ
HGLP-n	89.52(±0.52)	97.37(±0.46)	93.18(±0.70)	92.55(±0.66)	93.02(±0.35)	85.34(±0.70)	98.12(±0.17)
HGLP-e	90.01(±0.54)	97.70(±0.45)	93.49(±0.58)	92.73(±0.56)	93.18(±0.41)	85.38(±0.77)	98.10(±0.17)
HGLP	**90.13**(±0.57)	**97.73**(±0.40)	**93.64**(±0.59)	**92.86**(±0.63)	**93.29**(±0.42)	**85.57**(±0.75)	**98.17**(±0.17)

4 结束语

本文提出了一种基于 HGCN 的链路预测方法，针对现有方法时间和空间复杂过高和忽略节点间高阶关系的问题，使用时间和空间复杂度更低的 DHT 对原始的图结构进行处理以减少运行时间和内存占用，同时利用 HGCN 进行链路预测以学习高阶特征，有效提升了链路预测的精度。实验结果表明，基于 2 个常用评价指标 AUC 和 AP，本文提出的方法在 7 种不同领域的真实数据集上的表现均优于目前常用方法，且运行时间短、内存占用率低，在提升链路预测精度的同时，效率也得到了极大改善。由于本文方法是基于静态图的，而现实生活中存在的往往是动态图数据，因此如何将超图理论应用在动态

图的链路预测上将是下一步的研究方向。

参考文献

[1] LI J S, PENG J H, LIU S X, et al. Link prediction in directed networks utilizing the role of reciprocal links[J]. IEEE Access, 2020, 8: 28668-28680.

[2] 刘春雨, 陈庆锋, 莫少聪, 等. 基于逻辑规则和图神经网络的知识图谱补全[J]. 计算机工程, 2025, 51(3): 131-143.
LIU C Y, CHEN Q F, MO S C, et al. Knowledge graph completion based on logical rules and graph neural network[J]. Computer Engineering, 2025, 51(3): 131-143. (in Chinese)

[3] 吴志强, 解庆, 李琳, 等. 基于多模态融合的图神经网络推荐算法[J]. 计算机工程, 2024, 50(1): 91-100.
WU Z Q, XIE Q, LI L, et al. Graph neural network recommendation algorithm based on multimodal fusion[J]. Computer Engineering, 2024, 50(1): 91-100. (in Chinese)

[4] ASIF N A, SARKER Y, CHAKRABORTTY R K, et al. Graph neural network: a comprehensive review on non-euclidean space[J]. IEEE Access, 2021, 9: 60588-60606.

[5] ZHOU J, CUI G Q, HU S D, et al. Graph neural networks: a review of methods and applications[J]. AI Open, 2020, 1: 57-81.

[6] WU Z H, PAN S R, CHEN F W, et al. A comprehensive survey on graph neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(1): 4-24.

[7] GORI M, MONFARDINI G, SCARSELLI F. A new model for learning in graph domains[C]//Proceedings of the 2005 IEEE International Joint Conference on Neural Networks. Piscataway, USA: IEEE Press, 2005: 729-734.

[8] ISLAM K, ARIDHI S, SMAÏL-TABBONE M. A comparative study of similarity-based and GNN-based link prediction approaches[EB/OL]. (2020-08-20)[2024-03-28]. https://arxiv.org/pdf/2008.08879.pdf.

[9] JIANG N, NING B, DONG J Y. A survey of GNN-based graph similarity learning[C]// Proceedings of the 8th International Conference on Image, Vision and Computing. Piscataway, USA: IEEE Press, 2023: 650-654.

[10] CAI L, LI J D, WANG J, et al. Line graph neural networks for link prediction[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(9): 5103-5113.

[11] KIPF T, WELLING M. Semi-supervised classification with graph convolutional networks[EB/OL]. (2017-02-22)[2023-04-02]. https://arxiv.org/pdf/1609.02907.pdf.

[12] BAI S, ZHANG F H, TORR P H S. Hypergraph convolution and hypergraph attention[EB/OL]. (2020-10-10)[2023-11-15]. https://arxiv.org/pdf/1901.08150.pdf.

[13] JO J, BAEK J, LEE S, et al. Edge representation learning with hypergraphs[EB/OL]. (2021-10-29)[2023-11-13]. https://arxiv.org/pdf/2106.15845.pdf.

[14] MOKHTARI S, SHAKIBIAN H. An efficient link prediction method using community structures[C]// Proceedings of the 12th International Conference on Information and Knowledge Technology. Piscataway, USA: IEEE Press, 2021: 174-177.

[15] FANG Z H, TAN S L, WANG Y N, et al. Elementary subgraph features for link prediction with neural networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(4):3822-3831.

[16] NEWMAN M E J. Clustering and preferential attachment in growing networks[J]. Physical Review E Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics, 2001, 64(2 pt 2):025102.

[17] JACCARD P. Etude de la distribution florale dans une portion des alpes et du jura[J]. Bulletin de la Societe Vaudoise des Sciences Naturelles, 1901, 37: 547-579.

[18] XIE Y B, ZHOU T, WANG B H. Scale-free networks without growth[J]. Physica A: Statistical Mechanics and Its Applications, 2008, 387(7): 1683-1688.

[19] ZHOU T, LÜ L Y, ZHANG Y C. Predicting missing links via local information[J]. The European Physical Journal B, 2009, 71(4): 623-630.

[20] ADAMIC L A, ADAR E. Friends and neighbors on the web[J]. Social Networks, 2003, 25(3): 211-230.

[21] KATZ L. A new status index derived from sociometric analysis[J]. Psychometrika, 1953, 18(1): 39-43.

[22] JEH G, WIDOM J. SimRank: a measure of structural-context similarity[C]//Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2002: 538-543.

[23] BRIN S, PAGE L. Reprint of: the anatomy of a large-scale hypertextual Web search engine[J]. Computer Networks, 2012, 56(18): 3825-3833.

[24] ZHANG M H, CHEN Y X. Weisfeiler-Lehman neural machine for link prediction[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2017: 575-583.

[25] ZHANG M H, CHEN Y X. Link prediction based on graph neural networks[C]//Proceedings of International Conference on the Neural Information Processing Systems. New York, USA: ACM, 2018: 5165-5175.

[26] ZHANG M H, CUI Z C, NEUMANN M, et al. An end-to-end deep learning architecture for graph classification[C]// Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA: ACM, 2018: 4438-4445.

[27] LOUIS P, JACOB S A, SALEHI-ABARI A. Sampling enclosing subgraphs for link prediction[C]//Proceedings of the 31st ACM International Conference on Information and Knowledge Management. New York, USA: ACM, 2022: 4269-4273.

[28] GAO Y, FENG Y, JI S, et al. HGNN$^+$: general hypergraph neural networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(3): 3181-3199.

[29] GAO Y, ZHANG Z, LIN H, et al. Hypergraph learning: methods and practices[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(5): 2548-2566.

[30] WU H R, NG M K. Hypergraph convolution on nodes-hyperedges network for semi-supervised node classification[J]. ACM Transactions on Knowledge Discovery from Data, 2022, 16(4): 1-19.

[31] GARASUIE M M, SHABANKHAH M, KAMANDI A. Improving hypergraph attention and hypergraph convolution networks[C]// Proceedings of the 11th International Conference on Information and Knowledge Technology. Piscataway, USA: IEEE Press, 2020: 67-72.

[32] WATTS D J, STROGATZ S H. Collective dynamics of "small-world" networks[J]. Nature, 1998, 393(6684): 440-442.

[33] ROSSI R A, AHMED N K. The network data repository with interactive graph analytics and visualization[C]// Proceedings of the 29th AAAI Conference on Artificial Intelligence. New York, USA: ACM, 2015: 4292-4293.

[34] MERING C, KRAUSE R, SNEL B, et al. Comparative assessment of large-scale data sets of protein-protein interactions[J]. Nature, 2002, 417(6887): 399-403.

文字编辑 金胡考
栏目编辑 赖玉玲