

基于路径感知邻域的节点分类算法

郑文萍^{1,2,3}, 王晓敏¹, 韩兆荣¹

(1. 山西大学计算机与信息技术学院, 太原 030006; 2. 计算智能与中文信息处理教育部重点实验室(山西大学), 太原 030006; 3. 山西大学智能信息处理研究所, 太原 030006)

摘要: 图卷积神经网络通过将相似性高的邻居节点信息进行聚合以得到节点表示, 为节点选择合适邻域并进行有效聚合是图卷积网络的关键。现有的图卷积神经网络大多直接将多跳邻域内的节点信息聚合, 没有考虑到不同跳数邻域的聚合权重对网络中不同节点的差异性。针对此, 提出了一种基于路径感知邻域的节点分类算法(Path connectivity based neighbor-awareness node classification algorithm, PCNA), 通过网络中的路径连通信息确定节点邻域, 并自适应地感知不同长度路径对节点间相似性计算的影响权重, 指导图卷积神经网络的邻域聚合过程。PCNA由邻域感知器和节点分类器组成, 邻域感知器基于强化学习机制自适应地获取每个节点的聚合邻域及不同长度路径的影响权重, 再利用节点间的路径连通信息得到相似性矩阵; 节点分类器利用所得相似性矩阵进行邻域聚合得到节点表示, 并进行节点分类。在8个真实数据集上与10种经典算法的对比实验表明了所提算法在节点分类任务上有较好的性能。

关键词: 图卷积神经网络; 邻域聚合; 强化学习; 节点相似性; 节点分类

中图分类号: TP391 文献标志码: A

Path Connectivity Based Neighbor-Awareness Node Classification Algorithm

ZHENG Wenping^{1,2,3}, WANG Xiaomin¹, HAN Zhaorong¹

(1. College of Computer and Information Technology, Shanxi University, Taiyuan 030006, China; 2. Key Laboratory of Computation Intelligence and Chinese Information Processing(Shanxi University), Ministry of Education, Taiyuan 030006, China; 3. Institute of Intelligent Information Processing, Shanxi University, Taiyuan 030006, China)

Abstract: Graph convolutional neural networks obtain the node representation by aggregating the neighbor node information with high similarity, and selecting the appropriate neighborhood for the node and conducting effective aggregation are the keys to the graph convolutional networks. Most of the existing graph convolutional neural networks directly aggregate the node information in the multi-hop neighborhood, without considering the difference of the aggregation weights of different hop neighborhoods on different nodes in the network. Aiming at this, a path connectivity based neighbor-awareness node classification algorithm (PCNA) is proposed. The node neighborhood is determined by the path connectivity information in the network, and the influence weight of different length paths on the similarity calculation between nodes is adaptively perceived to guide the neighborhood aggregation process of graph convolutional neural network. Specifically, PCNA is composed of a neighborhood perceptron and a node classifier. The

基金项目: 国家自然科学基金(62072292); 山西省1331工程项目; 教育部产学研合作协同育人项目(220902842025336)。

收稿日期: 2024-06-23; 修订日期: 2024-08-11

neighborhood perceptron adaptively obtains the aggregated neighborhood of each node and the influence weights of paths with different lengths based on the reinforcement learning mechanism, and then uses the path connectivity information between nodes to obtain the similarity matrix. The node classifier uses the obtained similarity matrix to perform neighborhood aggregation to obtain node representation and classify nodes. The comparison experiments with 10 classical algorithms on eight real datasets show that the proposed algorithm has better performance in node classification tasks.

Key words: graph convolutional neural networks; neighborhood aggregation; reinforcement learning; node similarity; node classification

引 言

复杂网络是表示现实世界中数据对象之间关系的有力工具。节点分类根据拓扑结构或属性信息预测网络中节点的类别,被广泛应用于引文网络的论文分组^[1]、识别社交网络中的不良用户^[2]以及检测风控系统中的恶意账户^[3]等问题中。同一类节点通常具有较高的结构相似性和属性相似性,依据相似性可将具有相似性质的节点标记为相同的类别^[4]。基于图卷积的节点分类方法通过聚合节点的多跳邻域信息以得到节点低维空间嵌入,进而预测节点的类别。由于在网络中,两个节点越相似则越有可能产生连边,聚合节点的直接邻居和多跳邻居的信息可以使节点嵌入包含更多近邻节点信息,从而使相似节点在低维空间中表示也相近。目前多数图卷积网络没有考虑不同跳数的邻居为每个节点进行邻域聚合所提供的信息权重的差异,导致节点表示出现过平滑现象,影响节点分类性能。更准确地度量节点与其近邻相似性并将其用于节点的低维表示,有助于更准确地预测节点类别。

本文研究基于网络路径连通信息的节点相似性度量方法,根据节点与其近邻的路径连通情况,设计了一种图强化学习策略为每个节点选择合理的聚合邻域及确定不同跳数邻居的聚合权重,从而得到节点间的相似性矩阵。基于此,提出了一种基于路径感知邻域的节点分类算法(Path connectivity based neighbor-awareness node classification algorithm, PCNA),包括邻域感知器和节点分类器两部分。邻域感知器中设计了强化学习机制,自适应地获取每个节点的聚合邻域及不同长度路径的影响权重,再利用节点间的路径连通信息得到相似性矩阵。节点分类器中采用图卷积模块根据所得相似性矩阵进行邻域聚合得到节点表示,并预测节点类别。

本文主要贡献如下:(1) 设计了基于强化学习的节点邻域聚合策略,将不同长度路径对节点间相似性的影响权重确定问题建模为马尔可夫决策过程,为每个节点动态感知其聚合邻域及权重。(2) 提出了基于路径感知邻域的节点分类算法 PCNA,由邻域感知器和节点分类器组成。邻域感知器利用强化学习机制自适应地确定每个节点的聚合邻域及权重,得到基于节点间路径连通信息的相似性矩阵。节点分类器利用相似性矩阵聚合邻域,获得包含节点路径连通信息的节点表示,并执行节点分类。(3) 在 8 个真实数据集上与 10 种经典算法进行对比,结果表明邻域感知器学到的相似性矩阵可以指导邻域聚合;与简单邻域聚合算法相比,PCNA 更合理地确定聚合邻域及聚合权重,其分类准确度和 F_1 分数分别提升了 6.68% 和 2.98%。

1 相关工作

1.1 基本概念

现实世界中的复杂系统可抽象为由节点集 V 和边集 E 组成的图 $G=(V,E)$ 。其中,节点集表示为 $V=\{v_1, v_2, \dots, v_n\}$, $E=\{(v_i, v_j)|1 \leq i, j \leq n\}$ 为边集。节点 v_i 的直接邻居的集合定义为 $N(v_i)=$

$\{v_j | (v_i, v_j) \in E\}$, 节点 v_i 的度可表示为 $k_{v_i} = |N(v_i)|$ 。

定义图 G 的邻接矩阵为 $A = (A_{ij})_{n \times n}$, 其第 i 行第 j 列上的元素 A_{ij} 定义为

$$A_{ij} = \begin{cases} 1 & (v_i, v_j) \in E \\ 0 & (v_i, v_j) \notin E \end{cases}$$

则 A 的 l 次幂 $A^l (l \geq 1)$ 中的元素 $A_{ij}^{(l)}$ 表示 G 中节点 v_i 和 v_j 间长度为 l 的路径数, $A_{ii}^{(l)}$ 为 G 中节点 v_i 的长度为 l 的回路总数。如果一条路径中的节点各不相同, 则称此路径为一条点不重复路径。令长度为 l 的点不重复路径矩阵 $\hat{A}^{(l)} = (\hat{A}_{ij}^{(l)})_{n \times n}$, 其第 i 行第 j 列上的元素 $\hat{A}_{ij}^{(l)}$ 表示 G 中节点 v_i 和 v_j 间长度为 l 的点不重复路径数^[5]。

图 $G = (V, E)$ 上节点分类任务是指利用给定标记节点 $V_l \subset V$ 的类别信息, 预测未标记节点 $V_u = V - V_l$ 的类别。本文使用的主要符号与定义见表 1。

1.2 节点相似性指标

已有的基于网络拓扑结构的节点相似性度量方法通常是基于公共邻居信息或基于路径信息。基于公共邻居信息的相似性度量以两个节点公共邻居数计算相似性, 如式 (1~3) 所示的公共邻居 (Common neighbor, CN) 相似性^[6]、Jaccard 指标^[7]、Salton 指标^[8]等。

$$S_{v_i, v_j}^{\text{CN}} = |N(v_i) \cap N(v_j)| \quad (1)$$

$$S_{v_i, v_j}^{\text{Jaccard}} = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|} \quad (2)$$

$$S_{v_i, v_j}^{\text{Salton}} = \frac{|N(v_i) \cap N(v_j)|}{\sqrt{|N(v_i)| \times |N(v_j)|}} \quad (3)$$

Adamic-Adar (AA) 指标^[9]和资源分配 (Resource allocation, RA) 指标^[10]认为节点间的相似性应与节点度负相关, 即有

$$S_{v_i, v_j}^{\text{AA}} = \sum_{v \in N(v_i) \cap N(v_j)} \frac{1}{\ln |N(v)|} \quad (4)$$

$$S_{v_i, v_j}^{\text{RA}} = \sum_{v \in N(v_i) \cap N(v_j)} \frac{1}{|N(v)|} \quad (5)$$

基于路径信息的相似性度量方法利用两个节点间的路径信息来计算相似性。Katz 指标^[11]利用节点间所有的可能路径信息计算相似性, 路径对相似性的影响随着路径长度的增加而衰减, 有

$$S^{\text{Katz}} = \beta A + \beta^2 A^2 + \beta^3 A^3 + \dots = (I - \beta A)^{-1} - I \quad (6)$$

式中: β 为权重衰减因子, 其值小于邻接矩阵的最大特征值的倒数; I 为单位矩阵。局部相似性 (Local similarity, LS) 指标^[12]仅利用长度不超过 3 的路径计算节点间相似性。郑文萍等^[5]为消除网络中的大度节点对相似性的影响, 提出了基于点不重复路径的基于局部路径的相似矩阵 (Similarity metrics based on local paths, SLP) 指标。

表 1 主要符号及定义

Table 1 Main symbols and definitions

符号	定义
V, E	图的节点集和边集
G	由节点集 V 和边集 E 构成的图
$A, \hat{A}^{(l)}, \hat{A}_{v_i}^{(l)}$	邻接矩阵、长度为 l 的点不重复路径矩阵及矩阵 $\hat{A}^{(l)}$ 中节点 v 的对应行
X, x_v	特征矩阵及节点 v 的特征向量
Y, \hat{Y}	节点的真实类别和节点的预测类别
State	状态的有限集合
Action	动作的有限集合
Prob	状态转移概率
Reward	奖励函数
S	节点相似性矩阵
$s, a, \text{Prob}_v, R(s, a)$	节点 v 的状态、动作、状态转移概率、奖励

获取节点间所有可能的路径计算成本高,而网络中节点间通常在局部范围内产生影响,因此许多工作仅利用限定长度的路径计算节点间相似性。尽管如此,由于网络中各节点邻域内拓扑连接差异很大,在度量节点间相似性时,不同的节点可能需要考虑的路径长度不同,且给定长度的路径对节点相似性的影响权重也有差异。以上给出的相似性度量指标无法灵活考虑网络中每个节点的邻域拓扑连接的个性化差异,进而确定不同长度的路径对每个节点的特殊影响。

1.3 图卷积神经网络

图卷积神经网络将多跳邻域节点看作相似节点,通过聚合节点多跳邻域信息得到节点表示,其计算过程为^[13]

$$\mathbf{h}_u^{(k+1)} = U^{(k+1)} \left(\mathbf{h}_u^{(k)}, \sum_{v \in N(u)} M^{(k+1)}(\mathbf{h}_u^{(k)}, \mathbf{h}_v^{(k)}, A_{uv}) \right) \quad (7)$$

式中: $\mathbf{h}_u^{(k)}$ 为节点 u 在第 k 层的表示; $M^{(k+1)}(\bullet)$ 通过聚合节点 u 的直接邻居信息得到节点的邻域表示,也称为邻域聚合函数; $U^{(k+1)}(\bullet)$ 利用节点表示和其邻域表示得到下一层节点表示,也称为节点更新函数。

早期的图卷积网络(Graph convolutional network, GCN)^[14]采用平均邻域聚合函数,等同看待邻域中节点的影响力。图注意力网络(Graph attention network, GAT)^[15]采用注意力聚合机制考虑了聚合过程中不同邻居信息的聚合权重。图采样网络(Graph sample and aggregate, GraphSAGE)^[16]通过采样得到节点的邻域结构,为每个节点采样相同规模的邻居节点进行邻域聚合。随着GCN层数的增加,最终得到的节点表示包含的邻居跳数也越多。简化图卷积(Simple graph convolution, SGC)网络^[17]通过连续去除非线性激活函数,并折叠连续层间的权重矩阵来解决GCN中不必要的复杂性和冗余计算,同时这些简化操作不会对模型在下游任务的性能产生负面影响。然而,这些网络很少考虑为每个节点选择个性化的邻域聚合范围,以及不同跳数对相似性影响的个性化差异。

1.4 基于强化学习的图分析

强化学习通过智能体与环境的交互学习最优策略,已被应用于确定邻域聚合层数^[18]、关键子图选取^[19]、图聚类社区扩展^[20-21]以及图结构去噪^[22-23]等复杂网络分析任务中,得到更好的节点表示,同时提升图神经网络模型的可解释性。

Sun等^[19]给出了强化池机制自适应地选择网络的关键子图。Zhang等^[20]定义生成器从已生成社区的边界中选择节点进行社区扩展,把生成社区为真的概率作为奖励优化生成器。Wu等^[21]通过去除社区中的节点或扩展社区边界中的节点来灵活改善社区结构。Wang等^[22]提出策略网络来判断邻居是否为信号邻居,从有限的邻居集合中寻找最优的邻居集合,去除噪声邻居。Xu等^[23]将添加缺失边和删除不正确边的过程转化为边序列的生成过程,去除网络中的噪声边。Lai等^[18]提出元策略框架,为每个节点选择恰当的卷积层数,实现了为每个节点选择不同邻域聚合范围的目的。然而,该方法没有考虑不同跳数邻域对于聚合权重的差异。针对此问题,本文基于强化学习为每个节点选择聚合邻域及聚合权重,提出了一种基于路径感知邻域的节点分类算法。

2 PCNA 算法

为了使同类节点在嵌入空间内表示接近,不同类节点表示远离,节点表示应尽可能多地包含同类节点信息,减少异类节点信息。在拓扑连接上连接紧密的节点倾向于属于同一类,因此节点的多跳邻域中包含了其类别归属信息。然而由于图结构的不平衡性,不同节点相同跳数邻域内的类别构成差异较大,有些节点邻域内可能包含单一类别,而有些节点邻域则可能包含了过多其他类别节点。为了在

节点表示过程中,尽可能充分聚合到同类节点信息,且避免聚合不同类节点的信息,需要自适应地为每个节点感知合适的邻域范围和聚合权重,从而提高节点表示质量。

基于以上分析,本文提出PCNA算法,利用点不重复路径信息感知节点的有效邻域及聚合权重。算法包括邻域感知器和节点分类器两部分。在邻域感知器部分,基于强化学习策略为每个节点感知邻域范围及聚合权重,进而得到相似性矩阵;节点分类器采用图卷积神经网络预测节点标签。图1给出了PCNA算法的框架图。图1中, $Q(s; \varphi)$ 是参数为 φ 的策略网络, q_i 为评估分数, L_Q 和 L_F 分别为训练邻域感知器和节点分类器的损失函数。

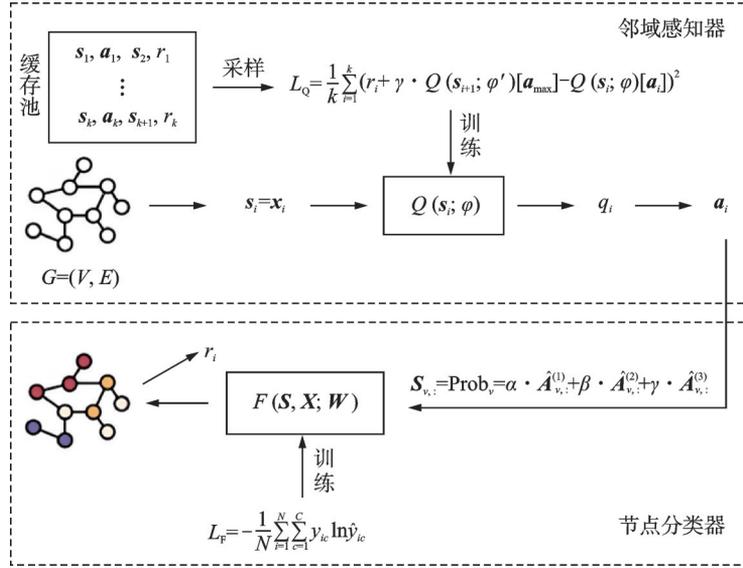


图1 PCNA算法框架

Fig.1 Framework of the PCNA algorithm

2.1 邻域感知器

邻域感知器确定节点聚合邻域跳数以及各跳邻域聚合权重,并将其建模为马尔可夫决策过程,利用强化学习来获取每个节点所需聚合的邻域跳数及不同跳数邻域的权重。

记图 $G=(V, E)$,其中 $V=\{v_1, v_2, \dots, v_n\}$ 是节点集, E 是边集, $n=|V|$, $A \in \mathbf{R}^{n \times n}$ 、 $D \in \mathbf{R}^{n \times n}$ 分别是 G 的邻接矩阵和度矩阵, V_{train} 、 V_{valid} 分别表示训练集和验证集中的节点。令 $M=(\text{State}, \text{Action}, \text{Prob}, \text{Reward})$ 为一个马尔可夫决策过程,其中 State 为状态的有限集合, Action 为动作的有限集合, $\text{Prob}: \text{State} \times \text{Action} \times \text{State} \rightarrow \mathbf{R}$ 表示状态 s 采取动作 a 后转移到下一状态 s' 的概率, $\text{Reward}: \text{State} \times \text{Action} \rightarrow \mathbf{R}$ 表示状态动作对 (s, a) 的奖励。状态、动作、状态转移概率、奖励函数的具体设计如下。

(1) 状态(State)为待采样节点的特征向量。

(2) 动作(Action)为不同跳数邻域重要性的所有可能分布,根据网络的小世界特性,此处将一个动作表示为三维向量 $a=(\alpha, \beta, \gamma)$,其中各分量在 $[0, 1]$ 区间取值,分别对应待采样节点一跳、二跳和三跳邻域的采样概率且满足 $\alpha + \beta + \gamma = 1$ 。

对待采样节点 v ,通过1个5层的多层感知机策略网络 $Q(s; \phi)$ 评估所有可能动作的分数 q , ϕ 是 Q 的可训练参数。再以随机探索概率 ϵ 从动作空间 Action 中随机选择动作,以 $1 - \epsilon$ 的概率选择 q 中分值最大的动作。此处设定停止动作为 $a_{\text{stop}}=(0, 0, 0)$,如果选择的动作是 $(\alpha, \beta, \gamma)=(0, 0, 0)$,则停止状态转移。

(3) 状态转移概率(Prob)。此处对当前节点 v 执行上一步得到的动作 $a = (\alpha, \beta, \gamma)$, 按照由式(8)定义的概率分布在网络中选取节点 u , 并将下一状态 s' 定义为 u 的特征向量。

$$\text{Prob}_v = \begin{cases} 1/n & a = a_{\text{stop}} \\ \alpha \cdot \hat{A}_{v,:}^{(1)} + \beta \cdot \hat{A}_{v,:}^{(2)} + \gamma \cdot \hat{A}_{v,:}^{(3)} & a \neq a_{\text{stop}} \end{cases} \quad (8)$$

式中: $\mathbf{1}$ 是 n 维的全 1 向量, n 是节点数, $\hat{A}^{(l)}$ 表示 G 中长度为 l 的点不重复路径矩阵, 其元素 $\hat{A}_{ij}^{(l)} = k$ 表示节点 v_i 和 v_j 间存在 k 条长度为 l 的点不重复路径, 其计算方式为

$$\hat{A}^{(1)} = A \quad (9)$$

$$\hat{A}^{(2)} = A^2 - D \quad (10)$$

$$\hat{A}^{(3)} = A^3 - DA - AD + A \quad (11)$$

$\hat{A}_{v,:}^{(l)}$ 是矩阵 $\hat{A}^{(l)}$ 中节点 v 的对应行, 表示 v 与图中节点的 l 跳连接的转移概率。 Prob_v 给出了由动作 $a = (\alpha, \beta, \gamma)$ 所定义的节点 v 的转移概率分布。当 $a = (0, 0, 0)$ 时从网络中随机选择下一状态节点; 否则按 $\alpha \cdot \hat{A}_{v,:}^{(1)} + \beta \cdot \hat{A}_{v,:}^{(2)} + \gamma \cdot \hat{A}_{v,:}^{(3)}$ 定义的概率分布选择下一状态的节点。

(4) 奖励函数(Reward)。利用验证集上分类性能计算奖励, 有

$$R(s, a) = \lambda \left(\delta(s, a) - \frac{\sum_{i=j-b}^{j-1} \delta(s_i, a_i)}{b} \right) \quad (12)$$

式中: b 表示计算奖励所考虑的历史经验数, λ 用于设置奖励函数的强度, $\delta(s, a)$ 是验证集上的分类性能。

2.2 节点分类器

采用一个简化的单层图卷积网络构建节点分类器 $F(\mathbf{S}, \mathbf{X}; \mathbf{W}) = \text{softmax}(\mathbf{X}\mathbf{S}\mathbf{W})$, $\mathbf{X} \in \mathbf{R}^{d \times n}$, $\mathbf{S} \in \mathbf{R}^{n \times n}$, d 为节点属性维度, \mathbf{W} 为一个可学习的权重矩阵。其基本原理是利用节点相似性矩阵 \mathbf{S} 对节点属性矩阵 \mathbf{X} 进行邻域聚合, 再采用 softmax 函数得到节点类归属概率, 将最大归属概率的类别作为预测的节点类别 \hat{Y} 。

相似性矩阵 \mathbf{S} 在算法执行过程中是即时更新的。 \mathbf{S} 被初始化为 1 个全 0 矩阵, 一旦使用策略网络 Q 计算了节点 v 的转移概率 Prob_v , 则用 Prob_v 替换 \mathbf{S} 中节点 v 的对应行。

设 $U \subseteq V$ 是 G 的一个节点子集, 采用节点分类器 $F(\mathbf{S}, \mathbf{X}; \mathbf{W})$ 预测 U 中节点的标签。首先利用当前策略网络 $Q(s; \phi)$ 为 U 中的所有节点计算对应动作, 并根据式(8)更新各节点 $v (v \in U)$ 的状态转移概率 $\text{Prob}_v \in \mathbf{R}^n$ 。组合所有节点的转移概率分布得到 U 中节点的相似性矩阵, 有

$$\mathbf{S} = [\text{Prob}_1, \text{Prob}_2, \dots, \text{Prob}_n] \quad (13)$$

2.3 模型优化

本文采用文献[24]提出的强化学习算法训练邻域感知器中的策略网络 $Q(s; \phi)$, 并通过最小化式(14)更新参数 ϕ , 即

$$L_Q = \frac{1}{k} \sum_{i=1}^k \left(r_i + \gamma \cdot Q(s_{i+1}; \phi') [a_{\max}] - Q(s_i; \phi) [a_i] \right)^2 \quad (14)$$

式中: $a_{\max} = \arg\max_a Q(s_{i+1}; \phi)$; ϕ' 为定期利用 ϕ 更新的参数; k 为批量大小; γ 为折扣系数。

通过最小化分类损失更新节点分类器 $F(\mathbf{S}, \mathbf{X}; \mathbf{W})$ 中的参数 \mathbf{W} , 即有

$$L_F = -\frac{1}{N} \sum_{i=1}^N \sum_{c=1}^C y_{ic} \ln \hat{y}_{ic} \quad (15)$$

式中： N 为训练样本数； C 为类别数； y_i 为节点 v_i 的真实类别，当类别 $y_i=c$ 时， $y_{ic}=1$ ，否则 $y_{ic}=0$ ； \hat{y}_{ic} 表示模型将样本 i 预测为类别 c 的概率。

2.4 算法描述

本文算法框架如算法 1 所示。

算法 1 基于路径感知邻域的节点分类算法

输入：图 $G=(V, E, X)$ ，策略网络训练回合数 P 、最大时间步数 T_1 ，分类器训练次数 T_2 ，随机探索概率 ϵ ，奖励参数 λ, b ，批量大小 k ，折扣系数 γ ，相似性矩阵 $S=0$ ，训练集 V_{train} ，验证集 V_{valid} ，测试集 V_{test} 。

输出：节点的预测类别 \hat{Y} 。

/*训练策略网络 $Q(s; \phi)$ */

(1) for $p=1, 2, \dots, P$ do

(2) $s_1 \leftarrow x_{\text{train}}$ /*将训练样本的特征作为初始状态*/;

(3) for $t=1, 2, \dots, T_1$ do

(4) 利用策略网络 $Q(s_t; \phi)$ 得到动作集合 a_t ;

(5) 对任一 $v \in s_t$:

(6) 根据式(8)得到 v 的状态转移概率 Prob_v ;

(7) 令 $S_{v,:} = \text{Prob}_v$;

(8) 按照 Prob_v 采样节点 w ，令 $s_{t+1} = s_t \cup \{w\}$;

(9) 在 V_{train} 上采用式(15)作为损失函数训练节点分类器 $F(S, X; W)$;

(10) 对任一 $v \in V_{\text{valid}}$:

(11) 利用策略网络 $Q(s_t; \phi)$ 得到 v 的对应动作 a ;

(12) 根据式(8)得到 v 的状态转移概率 Prob_v ;

(13) 令 $S_{v,:} = \text{Prob}_v$;

(14) 利用分类器 $F(S, X; W)$ 计算 V_{valid} 上节点的分类标签；并利用节点分类器通过式(12)在验证集上获得奖励 $R(s_t, a_t)$;

(15) 将经验 $(s_t, a_t, s_{t+1}, R(s_t, a_t))$ 存入缓存池；

(16) end for

(17) 从缓存池中采样 k 条经验，并根据式(14)优化策略网络 $Q(s; \phi)$;

(18) end for

/*训练分类器 $F(S, X; W)$ */

(19) for $t=1, 2, \dots, T_2$ do

(20) 对任一 $v \in V_{\text{train}}$:

(21) 利用策略网络得到 v 的对应动作 a ;

(22) 根据式(8)得到 v 的状态转移概率 Prob_v ;

(23) 令 $S_{v,:} = \text{Prob}_v$;

(24) 在 V_{train} 上采用式(15)作为损失函数优化节点分类器 $F(S, X; W)$;

(25) end for

/*输出测试集中节点类别*/

(26) 利用分类器 $F(S, X; W)$ 预测 V_{test} 的预测类别 \hat{Y} ;

(27) 结束。

在算法 1 中,第(1)~(18)行用来训练策略网络 Q ,在第 t 个时间步,将 $|V_{\text{train}}|$ 个节点输入策略网络 Q 计算动作,利用 $|V_{\text{valid}}|$ 个节点计算回报,时间代价为 $|V_{\text{train}} + V_{\text{valid}}|$,因此训练策略网络 Q 的总代价为 $O(T_1 \times P \times |V_{\text{train}} + V_{\text{valid}}|) < O(T_1 \times P \times |V|)$ 。第(19)~(25)行训练分类器,此处采用 SGC 作为分类器,每次迭代时间代价为 $O(|E|)^{[14]}$,因此,总代价为 $O(T_2 \times |E|)$ 。可以看出,尽管算法 PNCA 引入强化学习进行节点邻域采样,但训练策略网络 Q 的代价与网络节点数成线性关系。在稀疏网络中, $|V| \ll |E|$,因此训练策略网络 Q 并不会超过节点分类器的训练成本。第(26)~(27)行对测试集中节点标签进行预测,计算代价为 $O(|E|)$ 。综上所述,算法 1 的总训练时间复杂度和预测时间复杂度均与网络边数成线性相关关系,为 $O(|E|)$ 。

3 实验验证

3.1 实验设置

3.1.1 评价指标

本文采用准确度 ACC 和 F_1 分数对节点分类的性能进行评价,其计算方式分别为

$$ACC = \frac{TP + TN}{TP + FN + FP + TN} \tag{16}$$

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

式中:TP 为真正例数;TN 为真反例数;FP 为假正例数;FN 为假反例数;Precision = TP/(TP + FP) 为查准率;Recall = TP/(TP + FN) 为查全率。

3.1.2 数据集

本文在以下 8 个基准数据集上进行实验。

(1) Cora、Citeseer 和 Pubmed 是 3 个广泛使用的引文网络^[25-26]。在引文网络中,节点表示论文,边表示论文之间的引用关系。节点特征为论文的词袋表示,类别为论文的主题。

(2) Chameleon 和 Squirrel 是由维基百科中关于特定主题的网页构成的网络^[27]。节点表示网页,边表示网页之间的链接关系。节点特征为维基百科页面中的提供信息的名词。根据网页的月平均流量,将网页分为 5 类。

(3) Cornell、Texas 和 Wisconsin 三个数据来源于不同大学计算机系的网页数据集^[27]。其中,节点表示网页,边表示网页之间的超链接。节点特征为网页的词袋表示,根据网页所有者分为学生、项目、课程、职员和教职工等 5 类。

数据集的统计信息见表 2。对于引文网络,使用 Yang 等^[28]提供的标准数据集划分,即每类 20 个节

表 2 数据集的统计信息

Table 2 Statistical information of datasets

数据集	Cora	Citeseer	Pubmed	Chameleon	Squirrel	Cornell	Texas	Wisconsin
节点数	2 708	3 327	19 717	2 277	5 201	183	183	251
边数	5 429	4 732	44 338	36 101	217 073	295	309	499
特征数	1 433	3 703	500	2 325	2 089	1 703	1 703	1 703
类别数	7	6	3	5	5	5	5	5

点用于训练、500个节点用于验证、1 000个节点用于测试。对于其余网络,使用Pei等^[27]提供的随机划分,即训练集、验证集和测试集中节点数的占比分别为60%、20%和20%。

3.1.3 对比算法

为了探究节点特征与图结构对节点分类性能的影响,将本文所提算法PCNA与多层感知机(Multi-layer perceptron, MLP)、逻辑回归(Logistic regression, LR)及Node2vec^[29]进行对比。其中,MLP和LR只利用了节点特征,而Node2vec只利用了图结构。与传统图卷积神经网络,如SGC^[17]、GCN^[14]、GAT^[15]等进行对比,以验证本文邻域感知器的有效性。在基于强化学习的图分析方法中,可以从优化图神经网络层数的角度为每个节点构造不同层数的图神经网络,以聚合相应跳数邻域的信息,这可视作优化邻域聚合的一类方法^[18],如Policy-GNN。NAGphormer^[30]、LSGNN^[1]和GloGNN^[2]分别利用注意力机制、局部相似性和可学习的系数矩阵对节点不同邻域的信息进行加权融合,以从多跳邻域中学到有更多信息的节点表示。

3.1.4 实现细节

本文算法PCNA中的超参设置如下:批量大小 k 的取值范围为 $\{64, 128, 256, 512\}$,历史时间步数 b 的取值范围为 $\{0, 50, 100\}$,奖励强度 λ 的取值范围为 $\{0, 10, 100\}$,设置折扣系数 γ 为0.95,回合数和时间步数分别设置为20和5。利用Python3.7.16和CUDA11.4实现PCNA。在每个模型上运行10次,并计算平均值作为实验结果。

3.2 实验结果

表3给出了各算法在8个数据集上执行节点分类任务的准确度,其中加粗表示性能最好、加下划线的次之。可以看出,所提算法PCNA在所有数据集上的平均性能优于其他算法。具体地,PCNA比仅利用特征的方法(MLP和LR)以及仅利用图结构的方法(Node2vec)在分类准确度方面分别平均提升了26.28%、15.67%和27.9%。这表明,将图结构信息与属性信息相结合可以有效提高分类性能。相比于传统图卷积神经网络(SGC、GCN、GAT),PCNA在分类准确度方面分别平均提升了11.40%、12.06%和14.44%。这是由于PCNA在进行邻域聚合前加入了邻域感知器,能够自适应地学习不同节点所需的邻域聚合范围及不同跳数邻域的重要性;而传统图卷积神经网络对所有节点进行相同次数的图卷积操作,无法满足不同节点对聚合邻域跳数的需求。

表3 PCNA与基准模型的准确度比较

Table 3 Comparison of ACC between PCNA and baseline models

数据集	Cora	Citeseer	Pubmed	Chameleon	Squirrel	Cornell	Texas	Wisconsin	平均值
MLP	31.96	22.09	47.76	43.79	27.81	60.54	67.03	65.88	45.87
LR	57.60	61.4	72.8	37.32	27.93	55.14	67.30	72.35	56.48
Node2vec	62.08	40.78	56.12	47.00	31.29	34.59	45.68	36.47	44.25
SGC	80.70	68.30	77.70	<u>67.11</u>	48.32	48.65	54.05	41.18	60.75
GCN	80.76	<u>70.46</u>	78.75	59.65	42.91	52.70	52.70	42.75	60.09
GAT	82.31	69.91	77.15	55.44	36.48	45.95	52.70	41.76	57.71
Policy-GNN	67.74	61.59	<u>78.80</u>	58.45	<u>65.56</u>	81.62	75.64	59.87	68.66
NAGphormer	75.83	64.18	73.95	62.19	42.67	52.97	57.03	56.67	60.69
LSGNN	70.95	71.00	73.40	60.96	36.40	<u>78.38</u>	80.00	80.98	<u>69.01</u>
GloGNN	63.40	55.00	72.10	66.89	56.10	64.86	<u>78.38</u>	<u>80.39</u>	67.14
PCNA	<u>80.80</u>	68.40	79.30	74.78	67.44	62.16	75.68	68.63	72.15

与同样采用强化学习优化邻域聚合的算法 Policy-GNN 相比,PCNA 在平均分类准确度上仍然提升了 3.49%。这表明 PCNA 利用强化学习机制学习不同节点的聚合邻域跳数有助于得到更好的节点表示。可以看到,Policy-GNN 在 Cornell、Texas 和 Wisconsin 数据集上取得了较好的分类准确度,特别是在 Cornell 网络上优于 PCNA,这可能是因为 PCNA 通过采样获取节点邻域,在小规模网络上,节点度较小,通过采样获得的节点邻域信息较少,容易导致训练不足,而 Policy-GNN 选择节点 k 跳的所有邻居作为邻域,较 PCNA 获取的信息更充分。

图 2 显示了随着迭代次数的增加 PCNA 和 Policy-GNN 算法准确度的变化曲线,加粗曲线为 PCNA 的变化曲线。可以看出 PCNA 收敛所需的迭代次数较少,而 Policy-GNN 迭代次数较多且准确度波动较大。这是由于所提算法 PCNA 在邻域聚合中采用了基于点不重复路径的相似性矩阵,更容易获取节点间有效的多跳相互作用关系;而 Policy-GNN 通过增加网络层数聚合更多跳数邻域,引入了冗余信息,增加了计算代价。

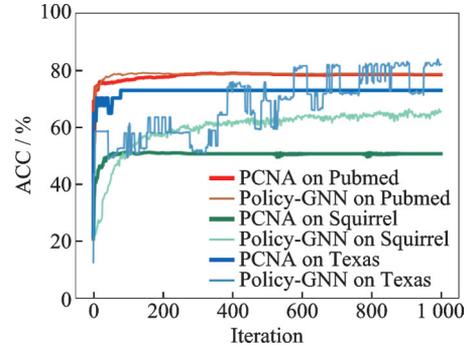


图 2 1 000 次迭代的准确度变化曲线

Fig.2 Accuracy change curves of 1 000 iterations

3.3 消融实验

为了验证邻域感知器中点不重复路径加权聚合策略的有效性,本节首先将 PCNA 的邻域选择策略替换为随机邻域选择(PCNA-random, PCNA-R)、基于注意力机制的邻域选择(PCNA-attention, PCNA-A)、基于节点余弦相似性的邻域选择(PCNA-cosine similarity, PCNA-C)和基于节点欧几里得相似性的邻域选择策略(PCNA-Euclidean similarity, PCNA-E)进行消融实验,以验证基于强化学习机制为节点学习个性化的邻域选择策略的有效性。同时,本节也与固定邻域选择策略进行比较,分别用 PCNA- i ($i=1,2,3$) 表示为每个节点仅选择其第 i 步邻域进行聚合。实验结果如表 4 和表 5 所示。其中,最后一行“+Impv”表示与性能最好的对比算法相比,PCNA 的提升率。可以看出,PCNA 在 8 个数据集上准确度和 F_1 分数平均提升分别为 6.68% 和 2.98%。这表明了本文算法利用强化学习所学的邻域聚合权重策略优于简单邻域聚合算法,说明了邻域感知器的有效性。图 3 可视化了邻域感知器采取的策略,横轴表示节点聚合邻域范围,纵轴表示聚合不同邻域跳数的节点所占百分比。可以观察到,不同类型数据集中的节点在邻域聚合时的邻域跳数分布不同,这表明 PCNA 利用强化学习为每个节点学习基于点不重复路径的邻域聚合权重策略的有效性。图 4 可视化了 PCNA 在 Pubmed 数据集上的分类结果,可以看出所划分的类别间有清晰的边界。

表 4 在 8 个数据集上的消融实验(准确度)

Table 4 Ablation experiments on eight datasets (ACC)

使用策略	数据集	Cora	Citeseer	Pubmed	Chameleon	Squirrel	Cornell	Texas	Wisconsin	平均值
固定邻域	PCNA-1	76.00	65.00	75.10	<u>67.11</u>	<u>48.32</u>	<u>48.65</u>	54.05	41.18	59.43
	PCNA-2	<u>80.70</u>	67.90	<u>77.70</u>	61.84	42.75	40.54	51.35	41.18	58.00
	PCNA-3	<u>80.70</u>	<u>68.30</u>	77.60	54.61	42.17	40.54	54.05	39.22	57.15
贪心策略	PCNA-A	18.98	18.91	36.02	20.55	19.66	41.89	40.27	33.92	28.78
	PCNA-C	66.96	57.09	32.16	37.48	23.77	38.38	47.30	39.22	42.80
	PCNA-E	59.09	62.55	50.07	36.91	25.17	41.62	51.89	50.20	47.19
随机策略	PCNA-R	71.48	64.95	75.32	53.09	43.24	44.32	<u>74.32</u>	<u>58.63</u>	<u>60.67</u>
本文算法	PCNA	80.80	68.40	79.30	74.78	67.44	62.16	75.68	68.63	72.15
+Impv		0.10	0.10	1.60	7.67	19.12	13.51	1.36	10.00	6.68

表5 在8个数据集上的消融实验(F_1 分数)Table 5 Ablation experiments on eight datasets (F_1 score)

%

使用策略	数据集	Cora	Citeseer	Pubmed	Chameleon	Squirrel	Cornell	Texas	Wisconsin	平均值
固定邻域	PCNA-1	74.38	61.93	75.03	<u>67.03</u>	<u>47.84</u>	32.50	23.23	27.58	51.19
	PCNA-2	79.82	65.00	<u>77.40</u>	61.90	41.53	26.82	19.00	24.73	49.53
	PCNA-3	80.00	<u>65.13</u>	77.27	54.48	40.09	26.61	27.33	23.90	49.35
贪心策略	PCNA-A	5.47	5.86	15.70	8.48	6.70	13.18	18.41	13.95	10.97
	PCNA-C	64.58	52.22	28.39	31.72	15.99	18.80	20.33	23.61	31.96
	PCNA-E	60.30	56.13	34.30	31.49	14.01	27.42	31.12	<u>33.04</u>	35.98
随机策略	PCNA-R	70.84	62.06	75.28	52.61	43.35	26.04	49.52	31.67	<u>51.42</u>
本文算法	PCNA	<u>79.90</u>	65.26	78.34	75.62	67.36	<u>30.01</u>	<u>42.59</u>	37.21	59.54
+Impv		-0.10	0.13	0.94	8.59	19.52	-2.49	-6.93	4.17	2.98

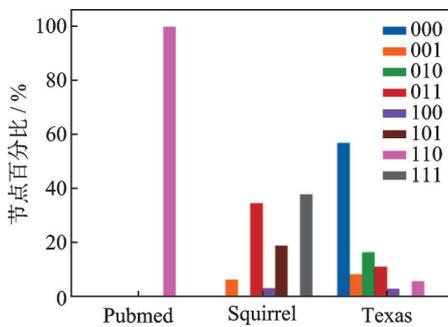


图3 聚合不同邻域跳数的节点所占百分比

Fig.3 Percentage of nodes aggregating different neighborhood ranges

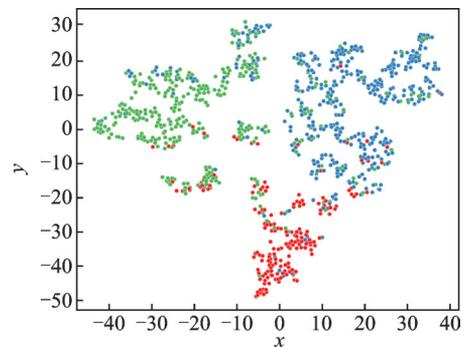


图4 Pubmed数据集上节点表示可视化

Fig.4 Node representation visualization on Pubmed dataset

3.4 动作空间设置

本节分析动作空间大小对算法性能的影响程度。考虑长度不超过3的点不重复路径,则所提算法PCNA中的一个动作可表示为3维向量 $\mathbf{a}=(a_1, a_2, a_3)$,分量 a_i 为学习到的 i 跳点不重复路径权重,在 $[0, 1]$ 区间取值。分别在 a_i 取值为 $\{0, 1\}$, $\{0, 0.33, 0.66, 1\}$, $\{0, 0.25, 0.5, 0.75, 1\}$, $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ 时构造动作空间。当 $a_i \in \{0, 1\}$ 时,取大小为8的动作空间 $\text{Action} = \{000, 001, 010, 011, 100, 101, 110, 111\}$ 。当 $a_i \in \{0, 0.33, 0.66, 1\}$ 时,可能的动作空间大小为64,为方便可视化,选择满足 $a_1 + a_2 + a_3 = 1$ 的动作进行实验,此时空间大小为11。对其他两种情况做类似处理,分别得到了大小为16和22的动作空间。图5给出了在不同动作空间大小下PCNA在实验数据集上的准确度。可以看到,在大多数数据集上, $a_i \in \{0, 1\}$ 时算法性能已达到最佳。

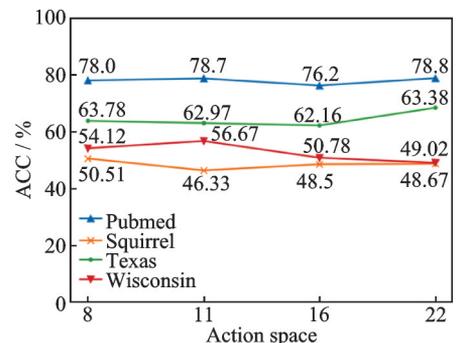


图5 动作空间对准确度的影响

Fig.5 Effect of action space on ACC

4 结束语

在图卷积操作前,通过确定每个节点聚合所需的邻域跳数及相应权重指导邻域聚合,能够提升模型的性能。本文提出了一种基于路径感知邻域的节点分类算法。在邻域感知器中,自适应地学习每个节点聚合的邻域跳数及不同跳数邻域的权重,并将其表达为节点相似性矩阵;然后在节点分类器中利用相似性矩阵进行邻域聚合,获得节点表示并进行节点分类。本文所提算法通过优化邻域聚合取得一定的提升。未来,将进一步研究算法对不同类型图数据的适应性和泛化能力,以扩展到更多图结构数据。此外,如何通过路径感知邻域聚合策略更好地理解图结构和节点间的相互作用是未来的研究方向。

参考文献:

- [1] CHEN Yuhan, LUO Yihong, TANG Jing, et al. Towards general graph neural network in node classification by local similarity [C]//Proceedings of the 32nd International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2023: 3550-3558.
- [2] LI Xiang, ZHU Renyu, CHENG Yao, et al. Finding global homophily in graph neural networks when meeting heterophily[C]//Proceedings of the 39th International Conference on Machine Learning. New York: ACM, 2022: 13242-13256.
- [3] CHEN Xuanze, HU Chenkai, GONG Shenbo, et al. Facilitating feature and topology lightweighting: An ethereum transaction graph compression method for malicious account detection[EB/OL]. (2024-05-14) [2024-16-15]. <https://arxiv.org/pdf/2405.08278>.
- [4] ZOU Minhao, GAN Zhongxue, CAO Ruizhi, et al. Similarity-navigated graph neural networks for node classification[J]. Information Sciences, 2023, 633: 41-69.
- [5] 郑文萍,车晨浩,钱宇华,等.一种基于节点间路径度量的图聚类算法[J].计算机学报,2020,43(7):1312-1327.
ZHENG Wenping, CHE Chenhao, QIAN Yuhua, et al. A graph clustering algorithm based on paths between nodes in complex networks[J]. Chinese Journal of Computers, 2020, 43(7): 1312-1327.
- [6] LYU Linyuan, ZHOU Tao. Link prediction in complex networks: A survey[J]. Physica A—Statistical Mechanics and Its Applications, 2011, 390(6): 1150-1170.
- [7] JACCARD P. Etude comparative de la distribution floraledansune portion des alpeset des jura[J]. Bulletin de la Societe Vaudoise des Sciences Naturelles, 1901, 37: 547-579.
- [8] 汪小帆,李翔,陈关荣.网络科学导论[M].北京:高等教育出版社,2012.
WANG Xiaofan, LI Xiang, CHEN Guanrong. Network science: An introduction[M]. Beijing: Higher Education Press, 2012.
- [9] ADAMIC L A, ADAR E. Friends and neighbors on the web[J]. Social Networks, 2003, 25(3): 211-230.
- [10] ZHOU Tao, LYU Linyuan, ZHANG Yicheng. Predicting missing links via local information[J]. The European Physical Journal B, 2009, 71(4): 623-630.
- [11] KATZ L. A new status index derived from sociometric analysis[J]. Psychometrika, 1953, 18(1): 39-43.
- [12] CHEN Zengqiang, XIE Zheng, ZHANG Qing. Community detection based on local topological information and its application in power grid[J]. Neurocomputing, 2015, 170: 384-392.
- [13] GILMER J, SCHOENHOLZ S, RILEY P, et al. Neural message passing for quantum chemistry[C]//Proceedings of the 34th International Conference on Machine Learning. New York: ACM, 2017: 1263-1272.
- [14] KIPF T, WELLING M. Semi-supervised classification with graph convolutional networks[C]//Proceedings of the 5th International Conference on Learning Representations. Washington DC: ICLR, 2017.
- [15] VELICKOVIC P, CUCURULL G, CASANOVA A, et al. Graph attention networks[C]//Proceedings of the 6th International Conference on Learning Representations. Washington DC: ICLR, 2018.
- [16] HAMILTON W, YING R, LESKOVEC J. Inductive representation learning on large graphs[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2017: 1024-1034.
- [17] WU F, ZHANG T, SOUZA A, et al. Simplifying graph convolutional networks[C]//Proceedings of the 36th International

- Conference on Machine Learning. New York: ACM, 2019: 6861-6871.
- [18] LAI K, ZHA D, ZHOU K, et al. Policy-GNN: Aggregation optimization for graph neural networks[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2020: 461-471.
- [19] SUN Qingyun, LI Jianxin, PENG Hao, et al. SUGAR: Subgraph neural network with reinforcement pooling and self-supervised mutual information mechanism[C]//Proceedings of the 30th International World Wide Web Conference. Berlin: Springer, 2021: 2081-2091.
- [20] ZHANG Yao, XIONG Yun, YE Yun, et al. SEAL: Learning heuristics for community detection with generative adversarial networks[C]//Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2020: 1103-1113.
- [21] WU Xixi, XIONG Yun, ZHANG Yao, et al. CLARE: A semi-supervised community detection algorithm[C]//Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2022: 2059-2069.
- [22] WANG Lu, YU Wenchao, WANG Wei, et al. Learning robust representations with graph denoising policy network[C]//Proceedings of the 19th IEEE International Conference on Data Mining. New York: IEEE, 2019: 1378-1383.
- [23] XU Jiarong, YANG Yang, PU Shiliang, et al. NetRL: Task-aware network denoising via deep reinforcement learning[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2023, 35 (1): 810-823.
- [24] HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double q-learning[C]//Proceedings of the 30th AAAI Conference on Artificial Intelligence. Palo Alto: AAAI, 2016: 2094-2100.
- [25] SEN P, NAMATA G, BILGIC M, et al. Collective classification in network data[J]. *AI Magazine*, 2008, 29 (3): 93-106.
- [26] NAMATA G, LONDON B, GETOOR L, et al. Query-driven active surveying for collective classification[C]//Proceedings of the 10th International Workshop on Mining and Learning with Graphs. New York: ACM, 2012.
- [27] PEI H, WEI B, CHANG K. C., et al. Geom-GCN: Geometric graph convolutional networks[C]//Proceedings of the 8th International Conference on Learning Representations. Washington DC: ICLR, 2020.
- [28] YANG Z L, COHEN W, SALAKHUTDINOV R. Revisiting semi-supervised learning with graph embeddings[C]//Proceedings of the 33rd International Conference on Machine Learning. New York: ACM, 2016: 40-48.
- [29] GROVER A, LESKOVEC J. Node2vec: Scalable feature learning for networks[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2016: 855-864.
- [30] CHEN Jinsong, GAO Kaiyuan, LI Gaichao, et al. NAGphormer: A tokenized graph transformer for node classification in large graphs[C]//Proceedings of the 11st International Conference on Learning Representations. Washington DC: ICLR, 2023.

作者简介:



郑文萍(1979-),通信作者,女,博士,教授,研究方向:复杂网络分析、生物信息学, E-mail: wpzheng@sxu.edu.cn。



王晓敏(1999-),女,硕士研究生,研究方向:复杂网络分析。



韩兆荣(1995-),男,博士研究生,研究方向:复杂网络分析。

(编辑:刘彦东)