



计算机科学

COMPUTER SCIENCE

随时间持续演化的流图神经网络

郭虎升, 张旭飞, 孙玉杰, 王文剑

引用本文

郭虎升, 张旭飞, 孙玉杰, 王文剑. 随时间持续演化的流图神经网络[J]. 计算机科学, 2025, 52(8): 118-126.

GUO Husheng, ZHANG Xufei, SUN Yujie, WANG Wenjian. [Continuously Evolution Streaming Graph Neural Network](#) [J]. Computer Science, 2025, 52(8): 118-126.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于双重分类和重建的跨域图异常检测](#)

Cross-domain Graph Anomaly Detection Via Dual Classification and Reconstruction

计算机科学, 2025, 52(8): 374-384. <https://doi.org/10.11896/jsjcx.241000140>

[基于动态阈值伪标签筛选的深度图对比聚类算法](#)

Deep Graph Contrastive Clustering Algorithm Based on Dynamic Threshold Pseudo-label Selection

计算机科学, 2025, 52(8): 100-108. <https://doi.org/10.11896/jsjcx.240700112>

[时空图神经网络在PM_{2.5}浓度预测中的应用综述](#)

Review on Application of Spatial-Temporal Graph Neural Network in PM_{2.5} Concentration Forecasting

计算机科学, 2025, 52(8): 71-85. <https://doi.org/10.11896/jsjcx.240700153>

[基于跨模态超图优化学习的多模态情感分析](#)

Cross-modal Hypergraph Optimisation Learning for Multimodal Sentiment Analysis

计算机科学, 2025, 52(7): 210-217. <https://doi.org/10.11896/jsjcx.240600127>

[基于脑网络连通性和脑电微状态的情感识别](#)

Emotion Recognition Based on Brain Network Connectivity and EEG Microstates

计算机科学, 2025, 52(7): 201-209. <https://doi.org/10.11896/jsjcx.240500087>

随时间持续演化的流图神经网络

郭虎升^{1,2} 张旭飞¹ 孙玉杰¹ 王文剑^{1,2}

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

(guohusheng@sxu.edu.cn)

摘要 流图在现实应用中广泛存在,其节点特征和结构特征会随时间推移而动态变化。尽管图神经网络在静态图节点分类中表现卓越,但其难以直接应用于流图,流图的持续演化会导致信息滞后和遗漏问题,所以模型难以准确提取流图特征。针对上述挑战,提出了一种随时间持续演化的流图神经网络(Continuously Evolution Streaming Graph Neural Network, CESGNN),以解决流图节点分类问题。该方法首先通过持续更新的图卷积网络(Continuous Updates Graph Convolutional Network, CU-GCN)增量地更新参数,以适应流图节点特征的变化,缓解信息滞后问题,然后自适应扩展的图神经网络(Adaptive Deepening Graph Neural Network, AD-GNN)通过将聚合和更新操作解耦,以挖掘流图深层特征,从而缓解信息遗漏问题。CESGNN通过有机地融合原始特征、CU-GCN提取的浅层特征和AD-GNN提取的深层特征,获得更准确、全面的流图特征表示。实验结果表明, CESGNN模型对流图具有良好的适应性和稳定性,提高了流图节点分类的准确率。

关键词: 流图;图神经网络;增量更新;聚合与更新解耦;特征融合

中图分类号 TP183

Continuously Evolution Streaming Graph Neural Network

GUO Husheng^{1,2}, ZHANG Xufei¹, SUN Yujie¹ and WANG Wenjian^{1,2}

1 School of Computer and Information Technology, Shanxi University, Taiyuan 030006, China

2 Key Laboratory of Computational Intelligence and Chinese Information Processing (Shanxi University), Taiyuan 030006, China

Abstract Streaming graphs are widely used in practical applications, and their node and structure characteristics change dynamically with time. Although Graph Neural Network (GNN) is excellent in static graph node classification, it is difficult to apply it directly to streaming graphs, because the continuous evolution of streaming graphs will lead to information lag and omission, it is difficult for models to accurately extract streaming graph features. To solve the above challenges, the Continuously Evolving Streaming Graph Neural Network (CESGNN) is proposed to solve the node classification problem of streaming graph. Firstly, the Continuous Updates Graph Convolutional Network (CU-GCN) incrementally updates parameters to adapt to changes in the node characteristics of the streaming graph to alleviate the information lag problem. Then the Adaptive Deepening Graph Neural Network (AD-GNN) alleviates the information omission problem by decoupling the aggregation and updates operations to dig deep features of the streaming graph. CESGNN organically combines the original features, the shallow features extracted by CU-GCN, and the deep features extracted by AD-GNN to obtain a more accurate and comprehensive representation of streaming graph features. The experimental results show that CESGNN model has good adaptability and stability of streaming graph, and improves the accuracy of node classification of streaming graph.

Keywords Streaming graphs, Graph Neural Networks, Incrementally updating, Decoupling aggregation and update, Feature fusion

1 引言

图是表示多种关系的一种结构,普遍存在于现实世界的

社交网络、交易网络中,并在社会网络分析^[1]、交易识别^[2]和流量预测^[3]等领域广泛应用。近年来,随着神经网络在多个领域取得成功并不断向图领域拓展,基于图结构的图神经

到稿日期:2024-10-31 返修日期:2025-05-25

基金项目:国家自然科学基金(62276157, U21A20513, 62476157, 62076154, 61503229);山西省重点研发计划(202202020101003)

This work was supported by the National Natural Science Foundation of China(62276157, U21A20513, 62476157, 62076154, 61503229) and Key Research & Development Program of Shanxi Province(202202020101003).

通信作者:王文剑(wjwang@sxu.edu.cn)

网络(Graph Neural Network, GNN)^[4-6]应运而生。这些网络结合了深度学习技术和图结构特征,可用于提取图特征表示,为图机器学习领域提供了丰富的工具,如节点分类^[7-8]、连接预测^[9-10]等。

尽管 GNN 在静态图的节点分类任务中取得了显著成果,但在现实问题中,图往往是动态的。会随着时间推移而发生变化^[11],静态 GNN 无法有效考虑流图的动态性质,导致其性能随时间推移逐渐降低^[12]。在流图演化过程中,导致 GNN 模型并不适用于流图的因素有两个。1)信息滞后:静态模型无法及时捕获流图节点特征的变化,模型在初始时刻进行训练和建模后模型参数不再进行更新,当流图节点的特征发生变化时,静态模型无法及时捕获这些变化。如图 1 所示, $t+1$ 时刻相比 t 时刻,节点的特征以及标签发生变化,但是静态模型仅适用初始时刻的节点特征,而无法适用节点特征和标签不断发生演化的流图。2)信息遗漏:静态模型无法灵活地根据当前的图结构聚合所需的节点特征信息。静态模型依赖于初始时刻的图结构,随时间推移,流图中节点之间的关系发生改变,导致其无法正确地聚合邻居节点的信息。如图 1 所示, $t+1$ 时刻相比 t 时刻添加了一条边,但是静态模型仅适用初始时刻的图结构,而无法适用图结构不断发生演化的流图。

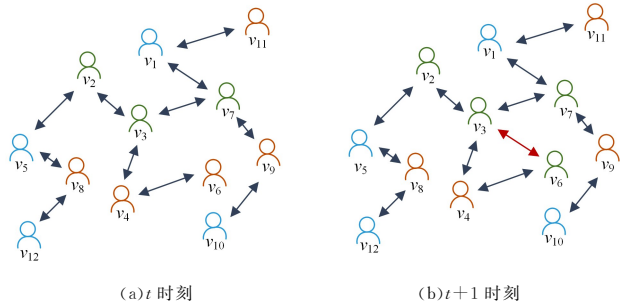


图 1 在 t 时刻和 $t+1$ 时刻流图结构特征

Fig. 1 Streaming graph structural features at time t and $t+1$

针对这些挑战,本文提出了随时间持续演化的流图神经网络(CESGNN),该模型包含了持续更新的图卷积网络模块(CU-GCN)与自适应扩展的图神经网络模块(AD-GNN)。其中 CU-GCN 是一个在线模型,通过持续学习流图节点特征变化来实时更新模型参数,以缓解信息滞后问题,从而提取图浅层特征。AD-GNN 则通过自适应增加模型深度来适应流图结构特征的变化,以缓解信息遗漏问题,从而挖掘图深层特征。然而,增加邻域聚合阶数会出现过平滑现象,AD-GNN 通过将聚合操作和更新操作解耦提高模型的灵活性,使模型在得以扩展的同时缓解了过平滑问题。最后, CESGNN 结合了流图原始特征以提高模型的收敛速度。本文将上述 3 种特征融合作为最终的流图特征表示。本文的主要贡献如下:

1)提出的 CU-GCN 方法通过不断地更新模型的参数使得模型能够捕获流图中节点特征和标签的变化。

2)提出的 AD-GNN 通过将聚合和更新操作解耦并增加模型聚合的邻居节点的深度,使得模型能够缓解信息遗漏

问题,并缓解过平滑现象。

3)在 3 个真实数据集上进行了实验,与其他基准模型进行了比较,验证了 CESGNN 模型的有效性和优越性。

2 相关工作

2.1 静态图节点分类

目前的静态图神经网络^[13]主要可以分为两大类:基于随机游走的和基于谱图理论的。基于随机游走的 GNN 利用随机游走来捕捉图中节点之间的局部和全局结构信息,并将其转换为节点嵌入向量,以此来表示节点的特征。例如 DeepWalk^[14], LINE^[15], Node2Vec^[16], Struc2Vec^[17] 和 Walklets^[18], 该类方法通常适用于较大规模的图,能够有效地捕获节点的结构信息。基于谱图理论的 GNN 利用图的邻接矩阵和节点特征进行信息传递和聚合。例如 GCN^[19], GraphSAGE^[12], GAT^[20] 和 GIN^[21], 这类方法在节点特征和图结构之间建立了更直接的联系,能够更好地利用节点的特征信息。这两类方法在静态图节点分类任务中发挥着重要作用,但并不适用于流图。

2.2 动态图节点分类

动态图节点分类^[22]通过捕捉时间信息来提升模型的表达力,该思路结合时间维度,使得模型能够捕捉到数据在不同时间点的演变和关联。DynamicTriad^[23]通过建模三元闭环结构来捕获网络的动态变化,从而在不同的时间点学习每个节点的向量表示;HTNE^[24]对时序数据和图数据进行联合建模,将 GCN^[19]与 peephole LSTM^[25]相结合捕获每个时间快照上的依赖性;EPNE^[26]使用了一种动态权重矩阵,该权重矩阵可以捕捉到流图中节点和边随时间的动态演化过程,并将这些动态信息传递到模型中;EvolveGCN^[27]使用循环神经网络(RNN)来演化 GCN 的参数,以捕获图序列的动态信息;DySAT^[28]通过使用自注意力机制来捕捉时间知识图谱中动态关系的特征;DyRep^[29]结合了网络结构的演化和节点间通信的动态变化,采用 RNN 进行参数化,以获得更丰富的网络表示;DynamicGCN^[30]是一种时序的图卷积神经网络模型,通过引入时间信息,将静态图卷积神经网络扩展为动态模型;DGNN^[31]在传统 GNN 中加入时间信息,可反映图数据的动态变化;ST-GCN^[32]通过在时间和空间上进行卷积来学习时空特征,将不同时间步的节点特征进行聚合及更新,捕捉节点在时间上的动态变化。由于这类动态神经网络的复杂性和难以训练的特点,其应用受到一定限制。本文针对持续变化的流图提出了随时间持续演化的流图神经网络,能够较好地适应流图的变化,提高流图节点分类的准确率。

3 持续演化的流图神经网络

为了解决流图中存在的信息滞后及信息遗漏问题,本文提出了一种随时间持续演化的流图神经网络(CESGNN)。该模型由 3 个模块组成:持续更新的图卷积网络(CU-GCN),自适应扩展的图神经网络(AD-GNN)以及特征融合模块。CESGNN 模型框架如图 2 所示,其中 CU-GCN 通过实时更新模型参数来适应流图节点特征的持续变化,从而缓解信息

滞后问题。然后 AD-GNN 采用一种增量的学习策略,根据图结构的改变实时调整网络参数,以适应流图结构的持续变化。最后通过特征融合模块及时准确地捕捉到每个节点的最新特征,从而为流图节点分类提供了更加精确的特征表示。

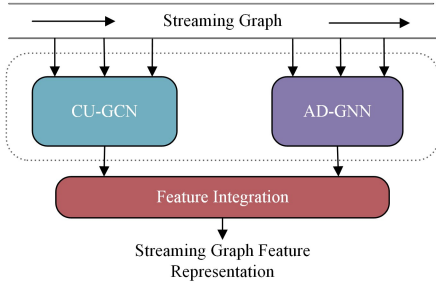


图2 CESGNN模型总体结构

Fig. 2 Overall structure of the CESGNN model

3.1 问题定义

本文中, $\mathcal{G} = (V, E, \mathbf{X})$ 表示一个无向属性图, $V = \{v_1, v_2, \dots, v_n\}$ 为节点的集合; $E \subseteq \{(v_i, v_j) \mid v_i, v_j \in V \text{ 且 } v_i \neq v_j\}$ 为边的集合; $\mathbf{X} \in \mathbb{R}^{n \times f}$ 表示节点特征, f 表示节点特征维度; $\mathbf{Y} \in \mathbb{R}^{n \times c}$ 表示节点标签, 其中 c 是节点标签个数; $\mathbf{A} \in \mathbb{R}^{n \times n}$ 表示邻接矩阵(若 i 与 j 相连通, 则 A_{ij} 为 1, 否则为 0); $\mathbf{D} \in \mathbb{R}^{n \times n}$ 表示度矩阵 ($D_{ii} = \sum_j A_{ij}$)。本文将流图视为多个时间步上的快照序列, 表示为 $S = \mathcal{G}^0, \mathcal{G}^1, \dots, \mathcal{G}^T$, T 是时间步的总数, 将流图在各时间步上初始时刻的状态作为当前时间步的快照 $\mathcal{G}^t = (V^t, E^t, \mathbf{X}^t)$, $V^t = V^{t+1}$, $|E^t| < |E^{t+1}|$ 。在本文中, 重点关注结构以及节点信息变化的情况, 即边的添加以及节点的特征和标签的变化。

3.2 持续更新的图卷积神经网络

图卷积网络 (Graph Convolutional Network, GCN) 通过聚合邻域节点特征来更新自身节点, 从而获得整个图的特征表示。GCN 将上一层节点的特征信息 \mathbf{Z}_{l-1} 作为输入, 执行聚合和更新操作得到当前层节点的特征信息 \mathbf{Z}_l 。为了更好地提取流图的特征表示, 本文通过持续学习的方式增量地更新 GCN 参数。该模型能够在初始快照图的基础上, 接收当前快照的节点特征进行持续更新, 这种方式使得模型能够适应流图中节点特征和标签的持续变化, 而无需重新训练。CU-GCN 在第 l 层的表示为:

$$\mathbf{Z}_l = \sigma((\tilde{\mathbf{D}}^0)^{-\frac{1}{2}} (\tilde{\mathbf{D}}^0 - \tilde{\mathbf{A}}^0) (\tilde{\mathbf{D}}^0)^{-\frac{1}{2}} \mathbf{Z}_{l-1} \mathbf{W}_l^t) \quad (1)$$

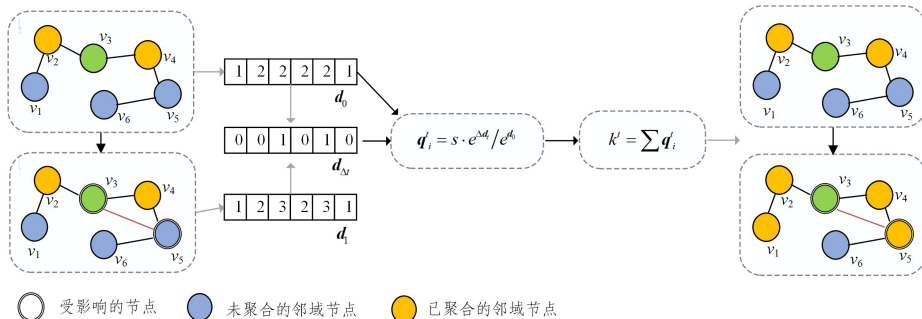


图3 确定邻域聚合阶数

Fig. 3 Determine the order of neighborhood aggregation

其中, $\mathbf{Z}_0 = \mathbf{X}^t$ 表示当前快照节点的初始特征, $\tilde{\mathbf{A}}^0 = \mathbf{A}^0 + \mathbf{I}$ 表示加入自循环后 \mathcal{G}^0 的邻接矩阵, $\tilde{\mathbf{D}}^0$ 表示 \mathcal{G}^0 的度矩阵, \mathbf{W}_l^t 表示第 l 层的节点参数矩阵, $\sigma(\cdot)$ 为非线性的激活函数。CU-GCN 经过两层的叠加, 得到当前快照浅层特征 $\mathbf{Z}_{\text{shallow}} = \mathbf{Z}_2$ 。在各时间步上, CU-GCN 通过交叉熵函数作为损失函数来训练模型, 损失函数如下:

$$\mathcal{L}_{\text{shallow}} = - \sum_i \sum_j \mathbf{Y}_{(i,j)}^t \ln(\mathbf{Z}_{\text{shallow}})_{(i,j)} \quad (2)$$

其中, \mathbf{Y}^t 表示为当前快照节点标签。

3.3 自适应拓展的图神经网络

自适应拓展的图神经网络 (AD-GNN) 模块能够根据不同快照之间的结构差异自适应地确定邻域聚合阶数, 从而灵活地调整模型的深度, 以挖掘流图的深层特征。

3.3.1 自适应确定邻域聚合阶数

本文提出了一种根据流图变化程度自适应地确定邻域聚合阶数的方法。假设不同快照节点数目是恒定的, 所以流图结构特征的动态性主要取决于边的变化。度矩阵含有图结构特征信息, 因此快照之间的差异可由度矩阵之间的差异量化。为简化计算, 本文定义度向量 $\mathbf{d}^t = [d_0^t, d_1^t, \dots, d_{n-1}^t]$, $d_i^t = D_{ii}^t$ 代替度矩阵进行运算。 $\nabla \mathbf{d}^t = |\mathbf{d}^t - \mathbf{d}^0|$ 表示当前快照 \mathcal{G}^t 和初始快照 \mathcal{G}^0 之间的差异。因此, 相对于初始快照, 当前快照中每个节点的变化程度可以由向量 \mathbf{q} 表示:

$$\begin{aligned} \mathbf{q}^t &= \mathbf{s}^t \cdot \frac{\mathbf{e}^{\nabla \mathbf{d}^t}}{\mathbf{e}^{\nabla \mathbf{d}^0}} \\ &= \mathbf{s}^t \cdot \left[\frac{\mathbf{e}^{(\nabla d^t)_0}}{\mathbf{e}^{(\nabla d^0)_0}}, \frac{\mathbf{e}^{(\nabla d^t)_1}}{\mathbf{e}^{(\nabla d^0)_1}}, \dots, \frac{\mathbf{e}^{(\nabla d^t)_{n-1}}}{\mathbf{e}^{(\nabla d^0)_{n-1}}} \right], \text{ where } \mathbf{s}^t = \frac{m_t}{n} \end{aligned} \quad (3)$$

其中, \mathbf{s}^t 是当前快照的稀疏因子, 用于反映图的稀疏程度。之后将各节点变化程度累加, 按式(4)所示得到当前快照的邻域聚合阶数 k^t 。

$$k^t = \alpha \cdot \frac{\sum_{i=0}^{n-1} \arctan(\mathbf{q}_i^t)}{n} \quad (4)$$

其中, α 是节点度改变的敏感因子, 是一个超参数。式(4)是根据当前快照中所有节点度的变化来确定邻域聚合的阶数, 并将局部变化的影响扩散到整个图中, 从而避免因局部节点的较大变化而对整体结果产生过度影响。除初始时间步外, 各时间步均会根据 \mathbf{d}^t 自适应地确定邻域聚合阶数。本节确定聚合邻域阶数的方法如图3所示。

3.3.2 解耦聚合更新操作加深模型

为了挖掘深层特征,本文通过解耦 GCN 模型中的聚合操作和更新操作以实现模型扩展,该方法有助于缓解模型过平滑现象^[33]。扩展模型挖掘深层特征主要分为 3 个步骤:1)计算各阶邻域节点的特征信息;2)计算各阶邻域节点特征信息的权重;3)将各阶邻域的特征与相应权重结合,得到当前节点的深层特征。根据 3.3.1 节中确定的邻域聚合阶数得到 k' 阶邻域节点的特征信息,如式(5)所示:

$$\mathbf{H}^l = [\mathbf{Z}_0^l, (\hat{\mathbf{A}}^l)^1 \mathbf{Z}_1^l, \dots, (\hat{\mathbf{A}}^l)^{k'} \mathbf{Z}_{k'}^l], \mathbf{Z}_0^l = \text{MLP}^1(\mathbf{X}^l) \quad (5)$$

其中, \mathbf{H}^l 为各阶邻域特征信息的序列, \mathbf{Z}_0^l 是自身节点特征, $(\hat{\mathbf{A}}^l)^i \mathbf{Z}_i^l$ 表示第 i 阶邻域节点的特征信息,融入了图结构信息,其中 $\hat{\mathbf{A}}^l = (\tilde{\mathbf{D}}^l)^{-\frac{1}{2}} \tilde{\mathbf{A}}^l (\tilde{\mathbf{D}}^l)^{-\frac{1}{2}}$, $\tilde{\mathbf{A}}^l = \mathbf{A}^l + \mathbf{I}$ 。各阶邻域节点的特征信息的权重通过式(6)自适应确定。

$$\mathbf{S}^l = \sigma(fc(\mathbf{H}_0^l, \mathbf{H}_{l+1}^l, \mathbf{H}_{l+2}^l, \dots, \mathbf{H}_{k'}^l)) \quad (6)$$

其中, l 表示 CU-GCN 的层数,由于浅层特征已经包含前 l 阶特征,所以不需要再次确定其权值。为了获取更准确各阶邻域节点的特征信息权重,需要在每个时间步上重新训练全连接层 fc 。最后,将权重序列与特征表示序列结合,得到深层特征 $\mathbf{Z}_{\text{deep}}^l$,如式(7)所示:

$$\mathbf{Z}_{\text{deep}}^l = \text{softmax}(\mathbf{S}_0^l \mathbf{H}_0^l + \sum_{i=l}^{k'} \mathbf{S}_i^l \mathbf{H}_i^l) \quad (7)$$

在计算 $\mathbf{Z}_{\text{deep}}^l$ 的过程中,原始特征 \mathbf{Z}_0^l 未被舍弃,有助于提高模型的收敛速度,从而加速整个模型的训练过程。挖掘深层特征的损失函数如式(8)所示:

$$\mathcal{L}_{\text{deep}}^l = - \sum_i \sum_j Y_{(i,j)}^l \ln(\mathbf{Z}_{\text{deep}}^l)_{(i,j)} \quad (8)$$

3.4 特征融合

本节将 CU-GCN 提取到的浅层特征与 AD-GNN 提取到的深层特征,以及不具有图结构的原始特征进行融合,作为流图特征表示。由于在初始时间步上没有深层特征,因此浅层特征便可作为流图特征表示。具体如式(9)所示:

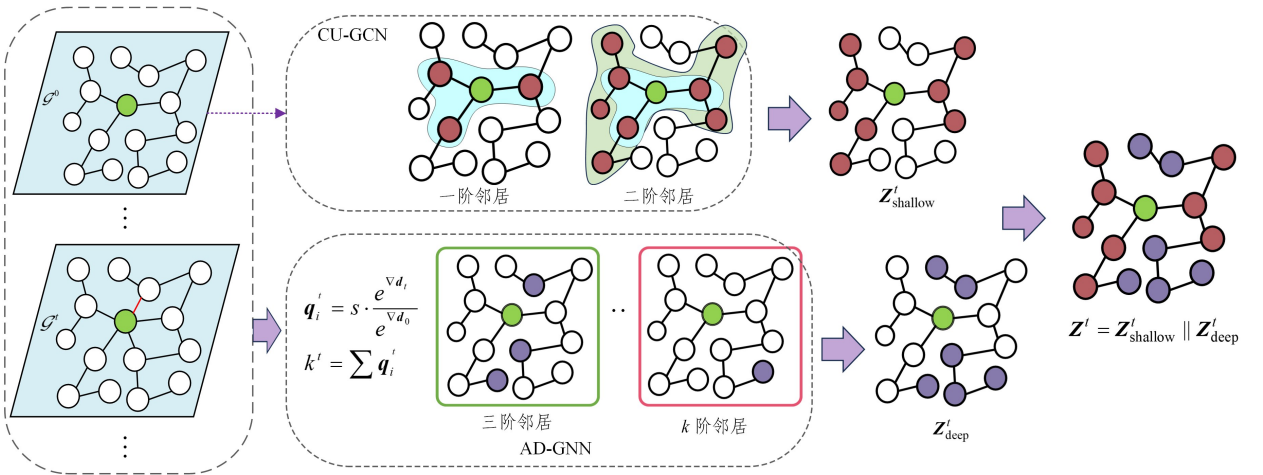


图4 随时间持续演化的流图神经网络

Fig. 4 Continuously evolution streaming graph neural network

算法1 CESGNN training algorithm

输入:各时间步上的快照 G^0, G^1, \dots, G^T ; 度向量 d^0, d^1, \dots, d^T ; 节点特征矩阵 X^0, X^1, \dots, X^T ; 节点标签矩阵 Y^0, Y^1, \dots, Y^T
输出:各时间步上的预测值 Z^0, Z^1, \dots, Z^T

$$\mathbf{Z}^0 = \mathbf{Z}_{\text{shallow}}^0 \quad (9)$$

$$\mathbf{Z}^l = \text{MLP}^2(\mathbf{Z}_0^l \parallel \mathbf{Z}_{\text{shallow}}^l \parallel \mathbf{Z}_{\text{deep}}^l)$$

其中, \parallel 为矩阵拼接操作,因此 $(\mathbf{Z}_0^l \parallel \mathbf{Z}_{\text{shallow}}^l \parallel \mathbf{Z}_{\text{deep}}^l) \in \mathbb{R}^{n \times 3f}$, 经过 MLP^2 将特征进行降维: $\mathbf{Z}^l \in \mathbb{R}^{n \times c}$ 。损失函数如式(10)所示:

$$\mathcal{L}^l = - \sum_i \sum_j Y_{(i,j)}^l \ln \mathbf{Z}_{(i,j)}^l \quad (10)$$

3.5 参数更新

CESGNN 是一种基于图神经网络的在线学习模型,能够在每个时间步上接收当前快照的结构特征及节点特征,持续学习流图的特征信息并增量地更新模型参数,以适应流图的动态演化。每个时间步都可以看作一个训练批次,模型参数的更新采用随机梯度下降(SGD)的方式进行,在各时间步均会更新模型参数。提取浅层特征模块中,可根据式(11)更新 CU-CGN 参数:

$$\mathbf{W}_{\text{CU-GCN}}^{l+1} = \mathbf{W}_{\text{CU-GCN}}^l - \gamma \nabla \mathcal{L}_{\text{shallow}}^l \quad (11)$$

挖掘深层特征模块中,可根据式(12)更新 AD-GNN 中 MLP^1 与 fc 参数:

$$(\mathbf{W}_{\text{MLP}^1}^{l+1}, \mathbf{W}_{fc}^{l+1}) = (\mathbf{W}_{\text{MLP}^1}^l, \mathbf{W}_{fc}^l) - \gamma \nabla \mathcal{L}_{\text{deep}}^l \quad (12)$$

特征融合模块中可根据式(13)更新 MLP^2 参数:

$$\mathbf{W}_{\text{MLP}^2}^{l+1} = \mathbf{W}_{\text{MLP}^2}^l - \gamma \nabla \mathcal{L}^l \quad (13)$$

其中,式(11)~式(13)中, \mathbf{W} 表示模型参数, γ 表示学习率。

CESGNN 的训练过程如算法 1 所示。在初始时间步,模型通过预训练得到初始快照并生成初始的图特征表示。在各时间步上,首先经过 CU-GCN 和 AD-GNN 得到当前快照的浅层特征和深层特征,然后经过特征融合生成每个节点的表示。在每个时间步结束时,模型将当前时间步得到的图特征表示用于下一个时间步的测试和训练。CESGNN 模型如图 4 所示。

1. let $\mathbf{Z}_0 = \mathbf{X}^0$
2. For $t=0, \dots, T$
3. 获得浅层特征: $\mathbf{Z}_{\text{shallow}}^0 = \mathbf{Z}_0^0$
4. If $t=0$:

5. $\mathbf{Z}^0 = \mathbf{Z}_{\text{shallow}}^0$
6. Continue;
7. End If
8. 更新 CU-GCN 模型参数 $\mathbf{W}_{\text{CU-GCN}}$
9. $\mathbf{q}^t = \mathbf{s} \cdot \frac{e^{\langle \mathbf{v}_d, \mathbf{z}_d \rangle}}{e^{\langle \mathbf{v}_d, \mathbf{z}_d \rangle_0}, \dots, \frac{e^{\langle \mathbf{v}_d, \mathbf{z}_d \rangle_{n-1}}}{e^{\langle \mathbf{v}_d, \mathbf{z}_d \rangle_{n-1}}}]$, $\mathbf{s} = \frac{\mathbf{m}_t}{n}$
10. $\mathbf{k}^t = \alpha \cdot \frac{\sum_{i=0}^{n-1} \arctan(\mathbf{q}_i^t)}{n}$
11. $\mathbf{H}^t = [\mathbf{Z}_0^t, \hat{\mathbf{A}} \mathbf{Z}_1^t, \dots, \hat{\mathbf{A}}^k \mathbf{Z}_k^t]$, $\mathbf{Z}_0^t = \text{MLP}^1(\mathbf{X}^t)$
12. $\mathbf{S}^t = \sigma(\text{fc}(\mathbf{H}_0^t, \mathbf{H}_{1+1}^t, \mathbf{H}_{1+2}^t, \dots, \mathbf{H}_k^t))$
13. $\mathbf{Z}_{\text{deep}}^t = \text{softmax}(\mathbf{S}_0^t \mathbf{H}_0^t + \sum_{i=1}^k \mathbf{S}_i^t \mathbf{H}_i^t)$
14. 更新 MLP^1 与 fc 参数
15. $\mathbf{Z}^t = \text{MLP}^2(\mathbf{Z}_0^t \parallel \mathbf{Z}_{\text{shallow}}^t \parallel \mathbf{Z}_{\text{deep}}^t)$
16. 更新 MLP^2 参数
17. End for

3.6 复杂度分析

CESGNN 模型计算效率高,操作简单,计算复杂度主要来自以下 3 个部分:1)CU-GCN 使用两层增量 GCN 提取浅层特征,计算复杂度为 $O(n^2)$,其中 n 为流图中节点个数;2)自适应确定邻域聚合阶数中对每个节点进行遍历,计算复杂度为 $O(n)$;3)AD-GNN 中计算当前快照的 $(\hat{\mathbf{A}}^t)^k \mathbf{Z}_k^t$,在计算过程中,如果采用从左到右的计算方法,复杂度为 $O(n^3)$,若采用从右到左的计算顺序,那么复杂度为 $O(cn^2)$,其中 c 为特征维度。因此总的的时间复杂度为 $O(cn^2) + O(n^2) + O(n) = O(n^2)$ 。

4 实验

本文通过对多个现实世界数据集进行节点分类任务^[34]的评估,且与目前流行的图特征表示方法进行比较,探究 CESGNN 模型在节点分类任务上的有效性与先进性。

4.1 数据集

本文的实验中使用了 3 个数据集:DeezerEurope, Facebook 和 Twitch。以上数据均采集自现实生活中的社交网络,并且能够体现出流图的演化过程(各时间步上快照均有边的增加,节点特征分布也会发生改变)。数据集的详细介绍如表 1 所列。

表 1 数据集信息

Table 1 Datasets information

数据集	节点	边	类别	特征
Facebook	22470	342004	4	128
DeezerEurope	28281	185504	2	128
Twitch	7126	70647	2	128

1)Facebook:该数据集是通过 Facebook Graph API 在 2017 年 11 月收集而来的一个包含验证的 Facebook 官方网页的无向图数据集,其节点代表 Facebook 官方网页,边代表网站之间的链接关系。数据集集中的每个节点都包含从站点描述中提取的节点特征。

2)DeezerEurope:该数据集于 2020 年 3 月从公共 API 中收集,是欧洲 Deezer 用户之间相互连接的社交网络。在这个数据集中,每个节点代表一个欧洲 Deezer 用户,每条边表

示两个用户之间的友好关系。每个节点还包括节点特征,这些特征包括用户的性别、年龄、地理位置和语言等信息。

3)Twitch:该数据集是一个 Twitch 社交网络,其中节点对应 Twitch 的用户,边代表用户之间的关系。每个节点还包括节点特征,这些特征包括用户的游戏喜好、位置和流媒体习惯等信息。

4.2 对比方法

将 CESGNN 模型与 GCN^[19], APPNP^[35], GraphSAGE^[12], ClusterGCN^[36], SGC^[37], TAGCN^[38], DySAT^[28], EvolveGCN^[27]和 ContinualGNN^[39]节点分类任务的经典模型进行比较。在线模型相比静态模型,可更好地适应流图的演化,故将静态模型设置为在线模型以验证 CESGNN 在流图节点分类问题上的有效性。

1)APPNP:该模型引入了个性化传播机制,该机制可以通过在每一层中将新的预测结果混合到前一层中,来提高模型的性能和稳定性。

2)GraphSAGE:该模型通过在节点邻域中采样和聚合信息来生成节点表示,采样策略可以根据应用场景进行配置,聚合策略可以选择不同的汇聚函数。

3)SGC:一种高效的图卷积网络模型,它使用一阶线性函数替代了传统的非线性激活函数,从而极大地简化了模型结构。

4)ClusterGCN:一种基于图划分的方法,将大型图划分成多个子图,在子图上进行 GCN 计算和信息更新,再将子图的节点表征聚合起来,更新全局节点表征。这种方法可以有效地加快计算速度。

5)TAGCN:一种自适应的图卷积网络模型,它通过学习每个节点在不同层中的邻域节点的重要性来调整每层的图卷积操作。

6)DySAT:一种动态图神经网络模型,它通过结构邻域和时间动态两个维度的联合自注意力来计算节点表示。

7)EvolveGCN:该模型通过使用循环神经网络适应时间维度,以调整图卷积网络(GCN)模型参数,从而捕获图序列的动态性。

8)ContinualGNN:一种基于持续学习的流图神经网络,通过从不断的增量训练过程中获取最新的节点表示。

4.3 实验设置

在本文所有实验中,CESGNN 中的 MLP 模型被设置为两层的全连接层,隐层包含 128 个神经元。对比方法均设置为两层的网络,隐藏层的节点个数均设置成 64。每个模型包含 200 个 epoch,学习率设置为 0.01,使用 Adam 优化器,实验结果是每种模型重复 5 次后的平均结果。

实验中为了有效缓解过拟合问题,本文将每个数据集中 60%的节点用作训练集,20%的节点用作验证集,其余 20%作为测试集,每个数据集分为 10 个时间步。DeezerEurope 数据集各时间步边的增幅为 4%;Facebook 数据集各时间步边的增幅为 1.5%;Twitch 数据集各时间步边的增幅为 4%。为进一步说明本文实验使用的数据集的动态性,本文将数据集随机分成 5 组,观察每组中节点的平均度数,如图 5 所示。

图 5 清晰地表达了随着时间步的增多,平均节点度数也

逐渐增加,即整个图愈加复杂,与现实相符。当流图随着时间变得复杂时,模型聚合邻域的阶数应该做出改变,因此在实验

过程中应该根据数据集将其设置为较为合理的网络结构改变敏感因子。

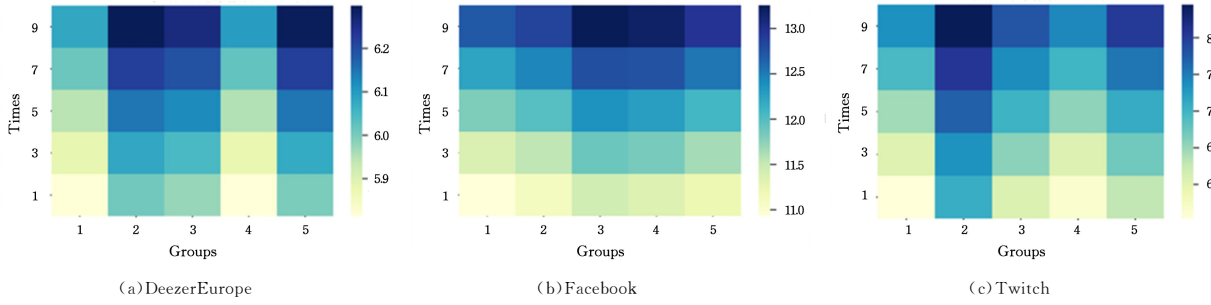


图5 不同时间步节点度数的变化

Fig. 5 Variation of the node degree at different time steps

4.4 自适应扩展模型

本节将展示流图演化过程中 CESGNN 在各时间快照上聚合的深度是如何变化的,并验证需要增加聚合深度的原因,从而说明 AD-GCN 模块的有效性。首先定义邻域节点数目的遗漏率为:

$$nl = 1 - \frac{N_{com}}{N_2} \quad (14)$$

其中, N_{com} 表示当前快照聚合到的邻域节点个数; N_2 表示最后快照聚合两层邻域节点的个数(GNN 模型常设置成两层网络)。以 Twitch 数据集为例可视化如何缓解信息遗漏问题,表 2 列出了一个节点不同邻域阶数相比最后快照的二阶邻域节点的特征信息下获取节点数目以及节点的损失率。可以发现,当聚合领域阶数低时,存在较严重的信息遗漏问题,通过增加领域聚合阶数可以有效缓解信息遗漏问题。当聚合邻居阶数为 5 时,节点损失率基本达到最小。图 6 对比了两个时间快照聚合的节点数目(其中红色为聚合节点),进一步说明了信息遗漏问题。AD-GCN 模块可以调整每个时间步聚合邻居的阶数,从而降低信息遗漏率。表 3 展示了在整个实验过程中各数据集邻域聚合阶数的变化。

表 2 不同邻域阶数下获取节点数目以及节点的损失率

Table 2 Number of nodes and loss rate of nodes are obtained under different neighborhood orders

邻域阶数	节点数目	损失率/%
2	1 058	44.29
3	4 767	8.20
4	6 163	3.40
5	6 369	3.20
6	6 394	3.70
7	6 396	3.70

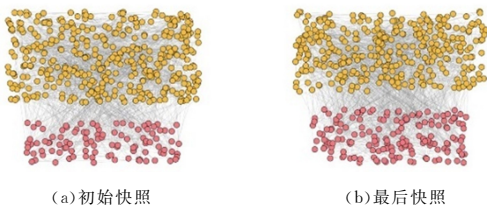


图 6 两个快照聚合的二阶邻域节点数目对比(电子版为彩图)
Fig. 6 Comparison of the number of second-order neighborhood nodes aggregated by two snapshots

表 3 各个时间步邻居阶数比较

Table 3 Comparison of neighbor orders at each time step

数据集	时间步									
	0	1	2	3	4	5	6	7	8	9
DeezerEurope	2	4	4	5	5	5	5	5	5	5
Facebook	2	5	5	5	6	6	6	7	7	7
Twitch	2	3	4	4	4	4	5	5	5	5

图 7 展示了不同聚合阶数下聚合到的节点数目,可以发现,当邻域聚合阶数增加到 3 时,节点聚合的邻居数量明显增加;而当邻域聚合阶数为 5 时,此时再增加邻域聚合阶数,虽然聚合的邻居数量依然会增加,但是节点损失率已经不再下降。这表明在当前快照下,聚合阶数为 5 时,可以解决信息遗漏问题。

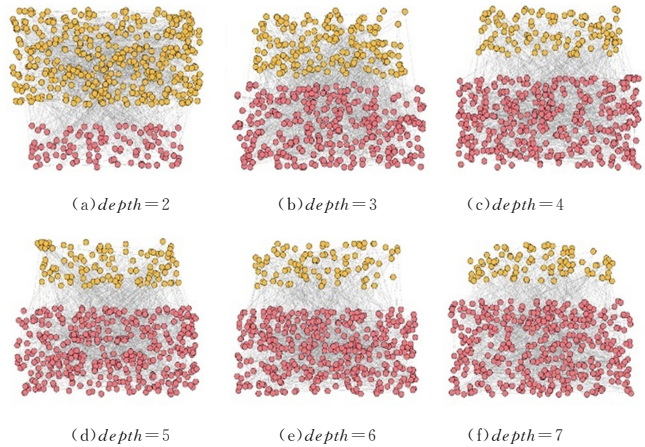


图 7 不同深度下 CESGNN 聚合到的邻域节点数

Fig. 7 Number of neighborhood nodes aggregated by CESGNN at different depths

通过分析,AD-GCN 可以通过持续扩展每个时间步聚合阶数,从而获得更多的邻域节点的特征信息,并且随着聚合深度的增加,节点损失率逐渐降低,由此说明本文提出的 AD-GCN 在流图上的有效性,其能够适应流图,提取更准确的流图特征表示。

4.5 实验结果

在各个数据集上进行节点分类实验,最终分类准确率和宏观 F1 值的比较如表 4 所列。

表 4 模型精度和 F1 指标

Table 4 Accuracy and F1 indicators of models

Model	DeezerEurope		Facebook		Twitch	
	Acc	F1	Acc	F1	Acc	F1
CESGNN	0.618±0.003	0.457±0.023	0.907±0.006	0.898±0.002	0.588±0.003	0.646±0.002
APPPNP	0.590±0.003	0.481±0.006	0.889±0.002	0.885±0.002	0.555±0.003	0.574±0.010
ClusterGCN	0.612±0.005	0.523±0.025	0.885±0.001	0.877±0.001	0.571±0.003	0.640±0.003
GCN	0.606±0.003	0.442±0.001	0.891±0.004	0.885±0.001	0.555±0.005	0.600±0.005
GraphSAGE	0.591±0.007	0.526±0.014	0.881±0.020	0.879±0.006	0.564±0.003	0.617±0.004
SGC	0.611±0.004	0.523±0.004	0.891±0.001	0.885±0.001	0.562±0.006	0.599±0.006
TAGCN	0.599±0.005	0.533±0.022	0.896±0.002	0.884±0.002	0.574±0.003	0.629±0.004
DySAT	0.613±0.005	0.577±0.003	0.893±0.005	0.889±0.007	0.577±0.005	0.570±0.004
EvolveGCN	0.614±0.004	0.580±0.010	0.896±0.004	0.888±0.005	0.579±0.007	0.572±0.005
ContinualGNN	0.612±0.006	0.578±0.010	0.898±0.007	0.892±0.007	0.580±0.008	0.636±0.008

在线模型能够持续接受流图节点特征并增量更新模型参数以缓解信息滞后问题,不仅如此,CESGNN 还能根据流图结构特征的变化程度自适应地确定邻域聚合阶数并扩展模型以缓解信息遗漏问题。相比其他方法,CESGNN 从两个方面缓解流图中存在的问题,因此能够更好地适应持续演化的流图。通过表 4 可以看到,本文方法在 DeezerEurope 数据集上

的 F1 值不是最高的,因为该数据集集中的节点特征和拓扑结构相对复杂,并且节点之间的相似度较高。本文方法在其余数据集的准确率以及 F1 均是最高,进一步说明了 CESGNN 模型的有效性。

图 8 展示了各时间步上 CESGNN 模型与其他对比模型的准确率。

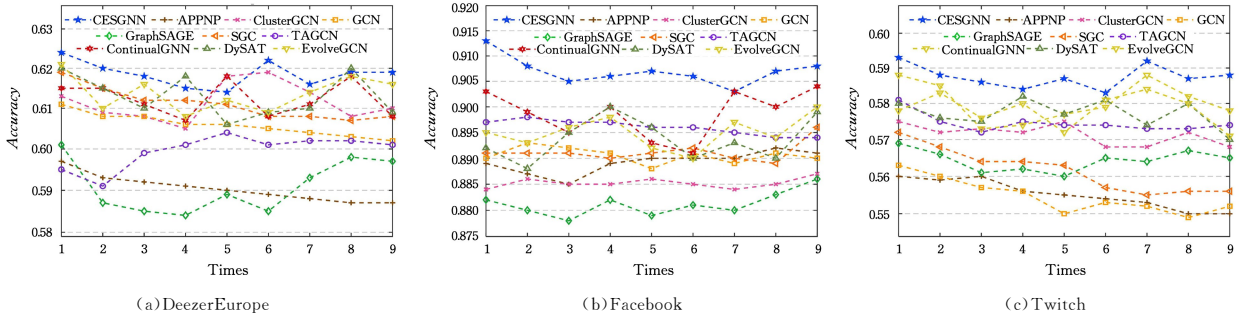


图 8 各数据集在不同时间快照的准确率

Fig. 8 Accuracy of each dataset at different time snapshots

根据图 8 的展示结果可以看出,CESGNN 在 Facebook 数据集和 Twitch 数据集的所有时间步上相比其他方法均取得了更好的准确率。这表明 CESGNN 模型具有更好的特征表达能力和更好的适应性,能够更好地捕捉流图中节点的变化。尽管在 DeezerEurope 数据集的第 5 个时间步上,CESGNN 模型的准确率略低于 ClusterGCN 方法,但总体来看,ClusterGCN 方法在整个过程中表现不稳定,而 CESGNN 模型的稳定性更高且平均准确率仍然是最优的。

通过观察图 9 可以发现,在各个时间步上,在 DeezerEurope 数据集上,CESGNN 的表现相对较差,仅高于 GCN 的 F1 值。值得注意的是,CESGNN 使用 GCN 提取网络特征作为浅层特征,而 GCN 在 DeezerEurope 数据集上的 F1 值最低,导致 CESGNN 表现不理想,但相比 GCN 仍有提高。CESGNN 在 Facebook 和 Twitch 数据集上的 F1 值依然最高,这进一步表明 CESGNN 在处理流图上依然具有较好的性能。

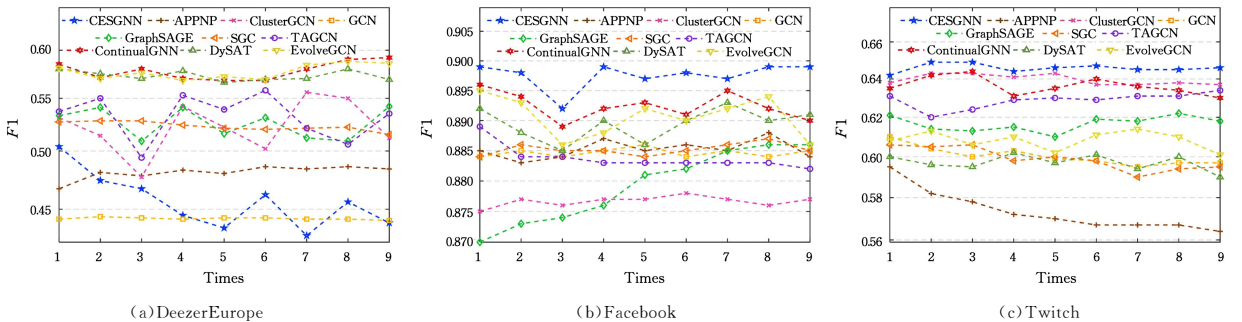


图 9 各数据集在不同时间快照的 F1

Fig. 9 F1 of each dataset at different time snapshots

4.6 邻域聚合阶数设置

CESGNN 模型通过增加邻域聚合的阶数来缓解信息

丢失的问题,不同的邻域聚合阶数避免信息丢失的能力是不同的,本节以 Twitch 数据集为例,深入讨论邻域聚合阶数的

设置。

通过表 3 可以看出,在 2 阶邻域聚合时,节点损失率非常大,随着邻域阶数的增加,节点损失率逐渐减小,但到 5 阶后节点损失率几乎不变。因此当聚合深度较小时,会有较大的信息遗漏问题,而当邻域阶数达到 5 阶时,损失率基本稳定,信息遗漏问题得以缓解,因此设置邻域阶数为 5。在式(6)中,参数 α 表示对图结构特征变化的敏感程度,在本节中设置了 5,7,9,11,13 这 5 个不同的数值进行对比。表 5 列出了 5 个不同参数下各时间步确定的邻域聚合阶数。

表 5 不同参数 α 下 Twitch 数据集邻域聚合阶数 S

Table 5 Twitch dataset neighborhood aggregation order for different S with different α

	时间步									
	0	1	2	3	4	5	6	7	8	9
$\alpha=5$	2	2	2	3	3	3	3	4	4	4
$\alpha=7$	2	2	3	3	3	4	4	4	4	5
$\alpha=9$	2	3	4	4	4	4	5	5	5	5
$\alpha=11$	2	4	4	4	4	5	5	5	6	6
$\alpha=13$	2	4	5	5	5	6	6	6	6	7

对不同参数下的 Twitch 数据集进行实验,观察其准确率,如图 10 所示。通过观察可以发现与之前的分析相符,当 α 较小时模型不能完全弥补信息丢失问题,导致性能不佳。然而,当邻域聚合阶数变大时,模型聚合的层次较深,虽然模型能够缓解过平滑问题,但却无法避免,因此其性能下降。

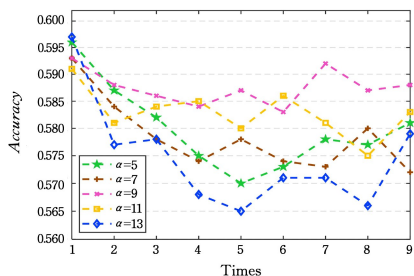


图 10 不同参数 α 下的模型结果

Fig. 10 Model results under different α

结束语 针对目前 GNN 在流图中存在信息滞后和信息遗漏问题,本文提出了一种随时间演化的流图神经网络。一方面,通过持续更新的图卷积网络使得模型能够适应流图中节点特征和标签的改变,从而缓解信息滞后问题;另一方面,通过聚合和更新操作解耦来增加模型聚合邻域的阶数,从而缓解信息遗漏问题。本文方法只能处理节点数目不变的情景,然而现实问题中普遍存在节点数目变化的数据,因此下一步工作将考虑适应节点数量变化的情况。

参考文献

[1] BRAMBILLA M, GASPARINI M. Brand community analysis on social networks using graph representation learning[C]// Proceedings of the 34th ACM SIGAPP Symposium on Applied Computing. 2019:2060-2069.

[2] ZHANG S, ZHOU D, YILDIRIM M Y, et al. Hidden: Hierarchical dense subgraph detection with application to financial fraud detection[C]// Proceedings of the 2017 SIAM International Conference on Data Mining. Society for Industrial and Applied

Mathematics. 2017:570-578.

[3] WANG X, MA Y, WANG Y, et al. Traffic flow prediction via spatial temporal graph neural network[C]// Proceedings of the Web Conference 2020. 2020:1082-1092.

[4] SCARSELLI F, GORI M, TSOI A C, et al. The graph neural network model[J]. IEEE Transactions on Neural Networks, 2008, 20(1):61-80.

[5] MICHELI A. Neural network for graphs: A contextual constructive approach[J]. IEEE Transactions on Neural Networks, 2009, 20(3):498-511.

[6] MA S, LIU J W, ZUO X. Survey on Graph Neural Network[J]. Journal of Computer Research and Development, 2022, 59(1):47-80.

[7] BHAGAT S, CORMODE G, MUTHUKRISHNAN S. Node classification in social networks[J]. Social Network Data Analytics, 2011:115-148.

[8] JI S Y, WEI Y X, DAI Q H, et al. Efficient hypergraph neural network on million-level data[J]. Scientia Sinica Informationis, 2024, 54(4):853-871.

[9] HASAN M A, ZAKI M J. A survey of link prediction in social networks[J]. Social Network Data Analytics, 2011:243-275.

[10] NI Q J, PENG W Q, ZHANG Z Z, et al. Spatial-Temporal Graph Neural Network for Traffic Flow Prediction Based on Information-Enhanced Transmission[J]. Journal of Computer Research and Development, 2022, 59(2):282-293.

[11] GUO J Y, LI R H, ZHANG Y, et al. Graph neural network based anomaly detection in dynamic networks[J]. Journal of Software, 2020, 31(3):748-762.

[12] HAMILTON W, YING Z, LESKOVEC J. Inductive representation learning on large graphs[C]// Proceedings of the Advances in Neural Information Processing Systems. 2017:1025-1035.

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

[14] PEROZZI B, AL-RFOU R, SKIENA S. Deepwalk: Online learning of social representations[C]// Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014:701-710.

[15] TANG J, QU M, WANG M, et al. Line: Large-scale information network embedding[C]// Proceedings of the 24th International Conference on World Wide Web. 2015:1067-1077.

[16] 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. 2016:855-864.

[17] RIBEIRO L F R, SAVERESE P H P, FIGUEIREDO D R. struc2vec: Learning node representations from structural identity[C]// Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017:385-394.

[18] PEROZZI B, KULKARNI V, CHEN H, et al. Don't walk, skip! online learning of multi-scale network embeddings[C]// Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017. 2017:258-265.

- [19] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[J]. arXiv:1609.02907, 2016.
- [20] VELIČKOVIĆ P, CUCURULL G, CASANOVA A, et al. Graph attention networks[C]// Proceedings of the International Conference on Learning Representations, 2018.
- [21] XU K, HU W, LESKOVEC J, et al. How powerful are graph neural networks? [C]// Proceeding of the International Conference on Learning Representations, 2019.
- [22] SKARDING J, GABRYŚ B, MUSIAL K. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey[J]. IEEE Access, 2021, 9: 79143-79168.
- [23] ZHOU L, YANG Y, REN X, et al. Dynamic network embedding by modeling triadic closure process [C] // Proceedings of the AAAI Conference on Artificial Intelligence, 2018; 571-578.
- [24] ZUO Y, LIU G, LIN H, et al. Embedding temporal network via neighborhood formation [C] // Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018; 2857-2866.
- [25] GERS F A, SCHRAUDOLPH N N, SCHMIDHUBER J. Learning precise timing with LSTM recurrent networks[J]. Journal of Machine Learning Research, 2002, 3(Aug): 115-143.
- [26] WANG J, JIN Y, SONG G, et al. EPNE: Evolutionary pattern preserving network embedding [C] // Proceedings of the 24th European Conference on Artificial Intelligence, 2020.
- [27] PAREJA A, DOMENICONI G, CHEN J, et al. Evolvegn: Evolving graph convolutional networks for dynamic graphs [C] // Proceedings of the AAAI Conference on Artificial Intelligence, 2020; 5363-5370.
- [28] SANKAR A, WU Y, GOU L, et al. Dysat: Deep neural representation learning on dynamic graphs via self-attention networks [C]// Proceedings of the 13th International Conference on Web Search and Data Mining, 2020; 519-527.
- [29] TRIVEDI R, FARAJTABAR M, BISWAL P, et al. Dyrep: Learning representations over dynamic graphs [C]// Proceedings of the International Conference on Learning Representations, 2019.
- [30] DENG S, RANGWALA H, NING Y. Learning dynamic context graphs for predicting social events [C]// Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019; 1007-1016.
- [31] MA Y, GUO Z, REN Z, et al. Streaming graph neural networks [C]// Proceedings of the 43rd international ACM SIGIR Conference on Research and Development in Information Retrieval, 2020; 719-728.
- [32] YAN S, XIONG Y, LIN D. Spatial temporal graph convolutional networks for skeleton-based action recognition [C] // Proceedings of the AAAI Conference on Artificial Intelligence, 2018.
- [33] LIU M, GAO H, JI S. Towards deeper graph neural networks [C]// Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2020; 338-348.
- [34] ZHANG T, LIAO B, YU J, et al. Benchmark testing and analysis of node classification tasks in graph neural networks [J]. Computer Science, 2024, 51(4): 132-150.
- [35] GASTEIGER J, BOJCHEVSKI A, GÜNNEMANN S. Predict then Propagate: Graph Neural Networks meet Personalized PageRank [C] // Proceedings of the International Conference on Learning Representations, 2018.
- [36] CHIANG W L, LIU X, SI S, et al. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks [C]// Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019; 257-266.
- [37] WU F, SOUZA A, ZHANG T, et al. Simplifying graph convolutional networks [C] // Proceedings of the International Conference on Machine Learning, 2019; 6861-6871.
- [38] DU J, ZHANG S H, WU G H, et al. Topology adaptive graph convolutional networks [J]. arXiv:1710.10370, 2017.
- [39] WANG J, SONG G, WU Y, et al. Streaming Graph Neural Networks via Continual Learning [C] // Proceedings of the 29th ACM International Conference on Information & Knowledge Management (CIKM'20), 2020; 1515-1524.



GUO Husheng, born in 1986, Ph.D, professor, Ph. D supervisor, is a senior member of CCF (No. 49571S). His main research interests include machine learning, data mining and computational intelligence.



WANG Wenjian, born in 1968. Ph. D, professor, Ph. D supervisor, is a outstanding member of CCF (No. 16143D). Her main research interests include machine learning, data mining and computational intelligence.

(责任编辑:何杨)