计算机科学
COMPUTER SCIENCE

基于启发式粗化算法的半监督图神经网络的训练加速框架及算法

陈裕丰, 黄增峰

引用本文

陈裕丰, 黄增峰. 基于启发式粗化算法的半监督图神经网络的训练加速框架及算法[J]. 计算机科学, 2024, 51(3): 48-55.

CHEN Yufeng , HUANG Zengfeng. Framework and Algorithms for Accelerating Training of Semi-supervised Graph Neural Network Based on Heuristic Coarsening Algorithms [J]. Computer Science, 2024, 51(3): 48-55.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.221200158

基于启发式粗化算法的半监督图神经网络的训练加速框架及算法

陈裕丰　　黄增峰
复旦大学大数据学院　上海 200433
(20210980146@fudan.edu.cn)

摘　要　图神经网络是当前阶段图机器学习的主流工具,发展势头强劲。通过构建抽象图结构,运用图神经网络模型能够高效地处理多种应用场景下的问题,包括节点预测、链接预测和图分类等方向。与之相对应,一直以来,在大规模图上的应用是图神经网络训练中的关键点和难点,如何有效、快速地在大规模图数据上进行图神经网络的训练和部署是阻碍图神经网络进一步工业化应用的一大难题。图神经网络因为能够利用图的网络结构的拓扑信息,所以在如节点预测的赛道上能够取得比一般其他神经网络如多层感知机等更好的效果,但是图的网络结构的节点个数和边的条数的规模增长制约了图神经网络的训练,真实数据集的节点数量规模达到千万级别甚至亿级别,或者是部分稠密的网络结构中边的数量规模亦达到了千万级别,使得传统的图神经网络训练方法均难以直接取得成效。针对以上问题,改进并提出了基于图粗化算法的新型图神经网络训练框架,并在此基础上提出了两种具体的训练算法,同时配合提出了两种简单的启发式图粗化算法。在精度损失可以接受和内存空间消耗大大降低的前提下,所提算法能够进一步显著地降低图神经网络的计算量,缩短训练时间,实验结果表明其在常见数据集上均能取得令人满意的成绩。

关键词:图神经网络;图粗化;训练加速;启发式;随机游走;无偏
中图分类号　TP391

Framework and Algorithms for Accelerating Training of Semi-supervised Graph Neural Network Based on Heuristic Coarsening Algorithms

CHEN Yufeng and HUANG Zengfeng
School of Data Science, Fudan University, Shanghai 200433, China

Abstract　Graph neural network is the mainstream tool of graph machine learning at the current stage, and it has broad development prospects. By constructing an abstract graph structure, the graph neural network model can be used to efficiently deal with problems in various application scenarios, including node prediction, link prediction, and graph classification. But the application on large-scale graphs has always been the key point and difficulty in graph neural network training. And how to effectively and quickly train and deploy graph neural networks on large-scale graph data is a major problem hindering the further industrial application of graph neural networks. Graph neural network can use the topological information of the network structure of the graph, so as to achieve better results than other general neural networks such as multi-layer perceptron on the node prediction problem. But the rapid growth of the number of nodes and edges of the graph's network structure restricts the training of the graph neural network, and the number of nodes in the real dataset is tens of millions or even billions, or the number of edges in some dense network structures has reached tens of millions. This makes it difficult for traditional graph neural network training methods to achieve direct results. This paper improves and proposes a new framework for graph neural network training based on heuristic graph coarsening algorithms, and proposes two specific training algorithms on this basis. Then this paper proposes two simple heuristic graph coarsening algorithms. Under the guarantee that the loss of accuracy is acceptable and the memory space consumption is greatly reduced, the proposed algorithm can further significantly reduce both calculation time and training time of graph neural networks. Experiment shows that satisfactory results can be achieved on common datasets.

Keywords　Graph neural network, Graph coarsening, Training acceleration, Heuristic, Random walk, Unbiased

近年来,图神经网络(Graph Neural Networks,GNN)已经成为图机器学习的主流工具,它在具有显式图结构或者隐式图结构的场景中有许多的应用,例如社交网络、推荐系统、生物蛋白质相互作用网络和知识图谱等方面,在实际应用中许多复杂的结构都可以用图进行结构的建模。

图神经网络的下游任务一般有 3 种,分别为节点分类(Node Classification),链接预测(Link Prediction)和图分类(Graph Classification);输入一般包括图结构信息,即节点和

到稿日期:2022-12-27　返修日期:2023-05-25
通信作者:黄增峰(huangzf@fudan.edu.cn)

节点之间的连边关系(Connect)、每个节点所对应的特征(Feature)和其他信息如节点标签(Label)等。节点预测旨在通过给定的图结构信息和部分节点特征以及类别标签，预测部分未给出的节点的所属类别；链接预测旨在通过给定的图结构信息和部分节点特征，预测目标节点之间是否存在相连接的边；图分类旨在通过给定的图结构信息和图特征以及标签，预测目标图的所属类别。

本文主要讨论的是节点分类这一最重要的赛道中的问题，且为半监督(Semi-supervised)学习。半监督学习的节点分类旨在利用部分给出的已标注节点的信息，学习节点的网络语义表示，对未标注的节点的类别进行预测。如在引文网络中，每一个节点表示一篇文章，每一条边表示其端点对应的文章之间的引用关系，节点特征一般由文章内容和标题通过语义分析处理后给出。特别地，因为在无向图(Undirected Graphs)中边不具备方向性，所以不对引用关系的主动和被动加以区分。

在传统的机器学习框架中，模型的损失函数(Loss Function)可以分解为单个样本的损失之和，因此可以使用小批次训练(Mini-batch)和随机优化(Stochastic Optimal)来处理比GPU内存大得多的训练集。

然而，在图神经网络的训练中，节点的表示从其邻居的表示递归计算得出，k层图神经网络中的每个样本的损失取决于其k阶(k hop)邻居，其所诱导的子图的大小随k呈指数级增长，使得上述的一般策略失效。因此，在早期的图神经网络的训练中，通常采用全批次(Full-batch)的梯度下降，这就是处理大规模图数据时所面临的问题。

本文改进并提出了一种新的基于图粗化(Graph Coarsening)算法的图神经网络训练框架，并在此基础上提出了两种具体的算法，同时配合提出了两种图粗化算法。在3个常用的公开数据集上，对本文算法和常见的图卷积神经网络节点分类算法进行了对比实验，实验结果在第5章进行说明。分析结果表明：在精度损失不大的前提下，相比非粗化算法，本文算法能够大大降低内存的开销；相比粗化算法，本文算法进一步显著地缩短了图神经网络的计算时间和训练时间。

本文的贡献主要包括两个方面：

1)提出了一个新的图神经网络框架，包括两个具体的图神经网络结构，相比原始的模型结构有更加严谨的理论推导，且实验效果得到了提升。

2)提出了两种更简单的粗化算法，相比原始的粗化算法，计算时间更短但实验效果相差不显著。

1 相关工作

为了方便叙述，表1列出了本文中常用的符号，并解释了符号的含义。

当前解决图神经网络训练加速的主流思路是基于邻居采样(Neighbor Sampling，NS)技巧和子图采样(Subgraph Sampling，GS)技巧，存在一些经典的解决方案。但除此之外，亦有一些独特的技巧用来加速训练。本章将对已有的相关方向的工作做一个简要的介绍。

记$\widetilde{\boldsymbol{A}}=\boldsymbol{A}+\boldsymbol{I}$为添加自循环的邻接矩阵，$\widetilde{\boldsymbol{D}}$为$\widetilde{\boldsymbol{A}}$的度矩阵，$\hat{\boldsymbol{A}}=\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}$为正则化后的添加自循环的邻接矩阵。

Kipf等在文献[1]中提出了基础的图卷积神经网络(Graph Convolutional Neural Networks，GCN)模型，更新公式为：

$$\boldsymbol{H}^{(k+1)}=\mathrm{ReLU}(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(k)}\boldsymbol{\omega}^{(k)})\triangleq(\hat{\boldsymbol{A}}\boldsymbol{H}^{(k)})$$

这是一种直推式(Transductive)的学习，意味着我们需要在计算时将整个邻接矩阵$\boldsymbol{A}$输入模型(随着相关理论的完善，消息传播神经网络模型最终解决了这一难题)，进行“全局”的矩阵乘法计算，从而更新每一层隐变量的值，当节点个数增加时，矩阵乘法的计算复杂度呈幂次级增长，这样的开销是巨大的。

此前已有大量的工作研究了如何加速图神经网络的训练，尤其是在大规模图上的图神经网络训练，并提出了各种技术来提高图神经网络训练的可扩展性，下面是不同方向的解决方案。

表1 符号含义

Table 1 Symbol explanation

符号	含义
G	有权(*weighted*)无向图，原图
G'	原图粗化后得到的新图
V	顶点集(node set)
E	边集(edge set)
$\boldsymbol{A}$	图的邻接(adjacency)矩阵
$\boldsymbol{L}$	图的拉普拉斯(Laplacian)矩阵
$\boldsymbol{D}$	图的度(degree)矩阵
$d(i)$	节点i的出度(out degree)
$\boldsymbol{W}$	边的权重矩阵，大小与$\boldsymbol{A}$相同
$\boldsymbol{H}^{(k)}$	第k层神经网络的隐变量矩阵
$\boldsymbol{h}_u^{(k)}$	节点u在第k层神经网络中的隐变量
$\mathcal{N}(u)$	节点u的邻域(neighbourhood)
φ	原图与新图节点之间的粗化映射
$\boldsymbol{P}$	粗化映射诱导的粗化矩阵
$\varphi^{-1}(u)$	节点u在粗化映射下的原像集
$\boldsymbol{M}(i,j)$	矩阵$\boldsymbol{M}$的第i行第j列元素
$\boldsymbol{M}(i,:)$	矩阵$\boldsymbol{M}$的第i行所表示的向量
$\boldsymbol{M}(:,j)$	矩阵$\boldsymbol{M}$的第j列所表示的向量
$\boldsymbol{v}^{(k)}$	在第k层神经网络语境下的向量$\boldsymbol{v}$
$\boldsymbol{v}_u$	对于节点u而言的向量

1.1 邻居采样

经典的方向是解耦节点之间的相互依赖关系，从而缩小邻居消息聚合的感受域，将更新公式的计算范围从全部样本缩小到单个样本，以适应小批次训练的要求。

Hamilton等[2]首次采用了邻居采样技巧，提出了GraphSAGE(Graph Sample and Aggregate)模型，这是一种与直推式学习相对的归纳(Inductive)学习方法，更新公式为：

$$\begin{cases}\boldsymbol{h}_{\mathcal{N}(u)}^{(k)}=Aggregate(\boldsymbol{h}_v^{(k)}\mid\forall\boldsymbol{v}\in\mathcal{N}_u)\\ \boldsymbol{h}_u^{(k+1)}=Linear(concat(\boldsymbol{h}_u^{(k)},\boldsymbol{h}_{\mathcal{N}(u)}^{(k)}))\end{cases}$$

其中，*aggregate*表示聚合函数，*concat*表示拼接函数。

此时更新公式和GCN有所区别，GraphSAGE不再采用图卷积神经网络的框架，而是改用消息传播神经网络(Message Passing Neural Network，MPNN)的框架，其物理意义是从邻居节点进行消息传递的聚合(这与GCN是一致的)，且目前已经被证明是一种非常有效的策略。但是，注意GraphSAGE使用叠加k层神经网络的方法来增大感受野，最后仍

等价于在 k 跳邻居诱导的子图上做消息聚合。当 k 足够大时所诱导出的子图和原图将相同，也就是说其算法的时间复杂度并没有实质性的改进，仅做到了常数级别的优化。

为了弥补这个缺陷，Chen 等[3]提出了 Fast GCN 模型，与 GraphSAGE 不同，它直接限制了节点的邻居采样范围，使用分层采样（Layer-wise Sampling）的技巧，通过重要性采样（Importance Sampling）的方式，从所有节点中采样部分节点以构成固定的邻居采样域。

对于任意的节点 $u\in V$，其以 $q(u)\propto\hat{A}(:,u)^2$ 的概率分布采样出一共 t 个节点，得到邻居采样域 $\mathcal{N}=u_1,u_2,\cdots,u_t$，更新公式为：

$$\begin{cases}\boldsymbol{h}_{\mathcal{N},u}^{(k)}=\sum_{j=1}^{t}\dfrac{\hat{\boldsymbol{A}}(\boldsymbol{u},\boldsymbol{u}_j)}{q(\boldsymbol{u}_j)}\boldsymbol{h}_{u_j}^{(k)}\\ \boldsymbol{h}_u^{(k+1)}=Linear(concat(\boldsymbol{h}_u^{(k)},\boldsymbol{h}_{\mathcal{N},u}^{(k)}))\end{cases}$$

在一个小批次内 GraphSAGE 的每个样本节点的邻居集合是独立的，而 Fast GCN 的所有样本节点共享同一个邻居集合，因此能够把计算复杂度直接控制到线性级别。但需要注意，Fast GCN 有一个大的缺陷，当待处理的图是大而稀疏的（实际上真实数据都是如此），由该方法采样得到的相邻层的样本可能根本没有关联，导致无法学习。

Chen 等[4]则基于无偏（Unbiased）近似的思想，提出了 S-GCN 的模型。在 S-GCN 方法中，保存每个节点在每一层上的历史值 $\boldsymbol{h}_v^{(k-1)}$，将其作为其真实值 $\boldsymbol{h}_v^{(k)}$ 的一个合理近似，更新公式为：

$$\begin{cases}\hat{\boldsymbol{A}}_{uv}^{(k)}=\dfrac{\mathcal{N}(u)}{\mathcal{N}^{(k)}(u)}\hat{\boldsymbol{A}}_{uv}\boldsymbol{1}_{v\in\mathcal{N}^{(k)}(u)}\\ \boldsymbol{H}^{(k+1)}=Linear(\hat{\boldsymbol{A}}^{(k)}(\boldsymbol{H}^{(k)}-\boldsymbol{H}^{(k-1)})+\hat{\boldsymbol{A}}\boldsymbol{H}^{(k-1)})\end{cases}$$

此时 $\hat{\boldsymbol{A}}^{(k)}$ 为基于邻居采样对 $\hat{\boldsymbol{A}}$ 的无偏近似，Chen 等通过理论分析得到，上述更新公式是 GraphSAGE 更新公式的无偏近似，且方差比 GraphSAGE 中所采用的邻居采样方法更小。

1.2 子图采样

与邻居采样的方法不同，子图采样技巧是从原图上采样子图，然后在这个子图上做局部的、全批次的 GCN 训练。

这样做不仅解决了邻居采样的指数化问题，而且可以对子图直接进行并行处理，大大提高了效率；但是采样出的子图和原图存在偏差（Bias），直接进行模型的迁移是不合理的，而且在实践中，每次从一个大图执行随机采样需要对内存进行多次随机访问，这对 GPU 来说依旧是高负荷的。

Chang 等[5]提出了 Cluster GCN 方法。该方法使用聚类的思想，把原图划分为若干个小块进行训练，以实现子图采样。具体思路为使用图聚类算法把原图节点分成 $V=\{V_1,V_2,\cdots,V_c\}$，然后不放回地随机从 V 中抽取共 q 个类别构成顶点集 $V'=\{V_{t_1},V_{t_2},\cdots,V_{t_c}\}$，从而诱导生成子图 G'。

图聚类算法让相似的节点分在一起，使得类内的节点分布和原图的节点分布有偏差。为了解决子图采样带来的问题，Cluster GCN 在训练时同时抽取多个类别作为一个批次参与训练，对节点分布进行平衡。特别地，在训练 GCN 时 Cluster GCN 还配合使用了对角增强的标准化策略，以搭建更深层的网络，具体更新公式为：

$$\boldsymbol{H}^{(k+1)}=Linear((\hat{\boldsymbol{A}}+\lambda\mathrm{diag}(\hat{\boldsymbol{A}}))\boldsymbol{H}^{(k)})$$

其中，λ 为对角增强系数。

Zeng 等[6]提出了 GraphSAINT（Graph Sampling Based Inductive Learning Method）模型，其思路与 S-GCN 类似，它显式地考虑了子图采样对 GCN 计算带来的偏差，以保证采样后节点的聚合过程是无偏的，并且使采样带来的方差尽量小，具体更新公式为：

$$\boldsymbol{h}_u^{(k+1)}=\sum_{v\in\boldsymbol{v}}\left(\frac{\boldsymbol{p}_u}{\boldsymbol{p}_{uv}}\boldsymbol{1}_{\{v|u\}}\right)\hat{\boldsymbol{A}}_{uv}(\boldsymbol{\omega}^{(k)})^{\mathrm{T}}\boldsymbol{h}_v^{(k)}$$

其中，$\boldsymbol{p}_u$ 表示节点 u 的被采样的概率，$\boldsymbol{p}_{uv}$ 表示边 e_{ij} 的被采样的概率，$\boldsymbol{1}_{\{v|u\}}$ 表示在节点 u 被采样后节点 v 是否被采样的指示向量。通过定义不同的 $\boldsymbol{p}_u$ 和 $\boldsymbol{p}_{uv}$ 即可诱导出不同的采样方法，这也是其与 Chang 等提出的 Cluster GCN 的显著差别之一。

1.3 图简化（Graph Reduction，GR）

与子图采样的想法类似，图简化指在对原图进行简化后得到的新图上进行训练。这种方法是可行的，但需要选择合适的简化方法，满足在缩小图的规模的同时能够保留特定的关键属性，以便进行后续的处理和分析。

图简化的思路分为两种：1）边简化，即删除原图的边，在边上进行简化；2）点简化，即删除原图的点，在点上进行简化。后者亦被称为图粗化（Graph Coarsening，GC），这是一个合适的加速 GNN 训练的框架。Huang 等[7]提出了 Coarsen GCN 的训练框架。

经理论分析，Huang 等发现在谱聚类（Spectral Clustering）粗化后的新图上做的 APPNP（Approximate Personalized Propagation of Neural Predictions）训练，等价于在原图上做部分受限的 APPNP[8]训练。因此，在粗化后的图 G' 上训练，能够大大减少图神经网络训练所需的参数，缩短训练消耗时间和减少运行内存开销。具体将在第 2.3 节中进行介绍。

但是和子图采样方法一样，图粗化需要对数据进行预处理，时间开销和实验效果与粗化算法的选择有较大的关系。本文在此基础上，基于无偏估计的思路，提出了一种新的图粗化方法以及基于图粗化算法的新的图神经网络训练框架，旨在在精度损失最小的情况下得到最快的可迁移的训练方式。

2 研究背景

本章将详细介绍图粗化的算法背景，以及基于图粗化的图神经网络训练框架。

2.1 图简化理论

Loukas[9]提出了图简化的框架。考虑一般的情形，对于原图 $G=(V,E,W)$，有 $N=|V|$ 个节点和 $M=|E|$ 条边。假设图 G 满足对称性和连通性，记 $\boldsymbol{L}$ 是一个能表征 G 的结构的稀疏半正定矩阵，即满足当且仅当 $e_{ij}\neq0$ 时有 $\boldsymbol{L}(i,j)\neq0$。

令 $\boldsymbol{L}_0=\boldsymbol{L}$ 和 $\boldsymbol{x}_0=\boldsymbol{x}$，其中 $\boldsymbol{x}$ 是任意的 N 维向量。定义如下的递归方式：

$$\begin{cases}\boldsymbol{L}_l=\boldsymbol{P}_l^{\mp}\boldsymbol{L}_{l-1}\boldsymbol{P}_l^{+}\\ \boldsymbol{x}_l=\boldsymbol{P}_l\boldsymbol{x}_{l-1}\end{cases}$$

其中，$\boldsymbol{P}_l\in\mathbb{R}^{N_l\times N_{l-1}}$ 是一个列数比行数多的窄矩阵，$l=1,2,\cdots,c$ 是简化的层级，符号 $+$ 和 $\mp$ 表示矩阵的伪逆和伪逆的

转置；N_l表示第 l 层级的简化维度，满足$N_0=N$以及$N_c=n\ll N$。向量$\boldsymbol{x}_c$在原空间$\mathbb{R}^N$上的提升（Lifted）由递归式$\tilde{\boldsymbol{x}}_{l-1}=\boldsymbol{P}_l^+\tilde{\boldsymbol{x}}_l$定义，其中$\tilde{\boldsymbol{x}}_c=\boldsymbol{x}_c$。

图简化试图找到一系列简化矩阵$\boldsymbol{P}_l$，使得表征图 G 结构的矩阵$\boldsymbol{L}$在简化过程中保持性质不变，称作不变性。

除此不变性以外，一个好的（Well-defined）图简化还需要对于任意的向量 $\boldsymbol{x}$ 均满足$|\tilde{\boldsymbol{x}}_c-\boldsymbol{x}|<\varepsilon$。其物理含义为经过 c 次简化后得到的向量，在原空间的提升与原向量是足够接近的。图简化的目的即是寻找满足要求的一系列简化矩阵$\boldsymbol{P}_l$。

2.2 图粗化理论

粗化是简化的一种，旨在找到一系列满射映射$\varphi_l:V_{l-1}\to V_l$，由此映射φ_l诱导的变换矩阵$\boldsymbol{P}_l$即是上述简化框架中的矩阵$\boldsymbol{P}_l$，称作粗化矩阵。

考虑 $\boldsymbol{L}$ 是原图 G 的拉普拉斯矩阵，则 Loukas 在文献[8]中指出，当且仅当$\boldsymbol{P}^+$的所有非零元素相等时，有$\boldsymbol{L}_c=\boldsymbol{P}^{\mp}\boldsymbol{L}\boldsymbol{P}^+$是新图$G_c$的拉普拉斯矩阵。特别地，此时的图粗化被称为保持拉普拉斯不变性的粗化（Laplacian Consistent Coarsening，LCC）。

定义由粗化映射$\varphi_l:V_{l-1}\to V_l$诱导的粗化矩阵$\boldsymbol{P}_l$满足：

$$\boldsymbol{P}_l(r,i)=\begin{cases}\dfrac{1}{|\boldsymbol{V}_{l-1}^{(r)}|}, & \text{当}v_i\in\boldsymbol{V}_{l-1}^{(r)}\\ 0, & \text{当}v_i\notin\boldsymbol{V}_{l-1}^{(r)}\end{cases}$$

其中，$V_{l-1}^{(r)}=\{v\in V_{l-1}\mid\varphi_l(v)=v_r,v_r\in V_l\}$表示$V_{l-1}$中所有映射到$V_l$中节点$v_r$的原像集。

此时可以得到：

$$\boldsymbol{P}_l^+(r,i)=\begin{cases}1, & \text{当}v_i\in\boldsymbol{V}_{l-1}^{(r)}\\ 0, & \text{当}v_i\notin\boldsymbol{V}_{l-1}^{(r)}\end{cases}$$

在此框架下得到的权重矩阵更新公式为：

$$\boldsymbol{W}_l(r,t)=\sum_{v_i\in\boldsymbol{V}_{l-1}^{(r)}}\sum_{v_j\in\boldsymbol{V}_{l-1}^{(t)}}\boldsymbol{W}_{l-1}(i,j)$$

2.3 图粗化框架下神经网络的训练

给定一个原图 G，图粗化算法可以返回一个节点个数更少的新图 G'。注意到图神经网络的权重矩阵大小与节点规模无关，仅与特征维度和隐变量维度相关，因此 Huang 等[7]给出了基于图粗化的图神经网络训练框架，具体如算法 1 所示。

算法 1 基于图粗化算法的图神经网络训练

输入：图 G=(V,E)，特征矩阵 $\mathbf{X}$，标签 Y，图神经网络 $\mathrm{GNN_G(W)}$，损失函数 $\mathscr{L}$

输出：网络的最优参数$\mathbf{W}^*$

1. 使用图粗化算法（例如谱聚类粗化）把原图 G 粗化成图 G′，归一化粗化矩阵为$\hat{\mathbf{P}}$；
2. 计算粗化后对应的特征矩阵$\mathbf{X}'=\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{X}$和标签$\mathbf{Y}'=\arg\max(\hat{\mathbf{X}}^{\mathrm{T}}\mathbf{Y})$；
3. 计算损失函数 $\mathscr{L}(\mathrm{GNN_{G'}(W)}(\mathbf{X}'),\mathbf{Y}')$，得到最优参数$\mathbf{W}^*$；
4. 返回$\mathbf{W}^*$作为图神经网络 $\mathrm{GN\,N_G(W)}$的最优训练参数。

3 关于传播系数的理论分析

由前述论证可以发现，原始的基于图粗化算法的训练框架存在两个问题：1）图粗化算法的选择是从已有的方法中挑选，具体来说，Huang 等所选择的是 Loukas 在文献[9-10]中提出的算法，这些算法和下游的图神经网络训练是解耦的，存在优化目标不一致的问题；2）并未对消息传播框架做更进一步的分析，只是简单地套用了消息传播神经网络模型，缺乏对应性。本章基于第二点，详细论述了基于图粗化算法的无偏的图神经网络聚合方式。

3.1 前提条件

1）采用消息传播神经网络模型进行相应的特征计算：$\boldsymbol{h}_u^{(k+1)}=\sum_{v\in\mathcal{N}(u)}\alpha_{uv}\boldsymbol{h}_v^{(k)}$；

2）原图 G 和新图 G' 上的节点特征由粗化映射 φ 所联系：$\boldsymbol{h}_p=\dfrac{1}{\varphi^{-1}(p)}\sum_{r\in\varphi^{-1}(p)}\boldsymbol{h}_r$；

3）权重 $\boldsymbol{w}$ 依公式$\boldsymbol{w}_{pq}=\sum_{r\in\varphi^{-1}(p)}\sum_{s\in\varphi^{-1}(q)}\boldsymbol{w}_{rs}$进行转移。

我们希望在特征计算的层面，原图和新图上的消息传递公式是等价的，也就是说，在新图 G' 上做消息传递、计算特征、梯度回传等操作，等价于在原图上做。注意，此处为了简便，与 Wu 等在文献[11]中提出的假设一致，并不考虑包括激活函数在内的非线性变换部分。

考虑新图 G' 上的两层节点特征，分别是节点 p 在第$k+1$层的隐变量值$\boldsymbol{h}_p^{(k+1)}$和节点 q 在第 k 层的隐变量值$\boldsymbol{h}_q^{(k)}$，它们的节点特征均由粗化映射计算得出。

$$\begin{cases}\boldsymbol{h}_p^{(k+1)}=\dfrac{1}{\varphi^{-1}(p)}\sum_{r\in\varphi^{-1}(p)}\boldsymbol{h}_r^{(k+1)}\\ \boldsymbol{h}_q^{(k)}=\dfrac{1}{\varphi^{-1}(q)}\sum_{s\in\varphi^{-1}(q)}\boldsymbol{h}_s^{(k)}\end{cases}$$

希望对于原图 G 上的消息聚合，能够形式上对应于新图 G' 上的消息聚合，即：

$$\begin{cases}\boldsymbol{h}_p^{(k+1)}=\sum_{q\in\mathcal{N}(p)}\alpha_{pq}\boldsymbol{h}_q^{(k)}\\ \boldsymbol{h}_r^{(k+1)}=\sum_{s\in\mathcal{N}(r)}\alpha_{rs}\boldsymbol{h}_s^{(k)}\end{cases}$$

应同时成立。

如图 1 所示，直观上分析，本文的研究目标是 $f(\varphi(G))$和$\varphi(f(G))$是否相等，即粗化算法和图神经网络是否具备可交换性。我们将上述表示如下论断，记为原命题 A。

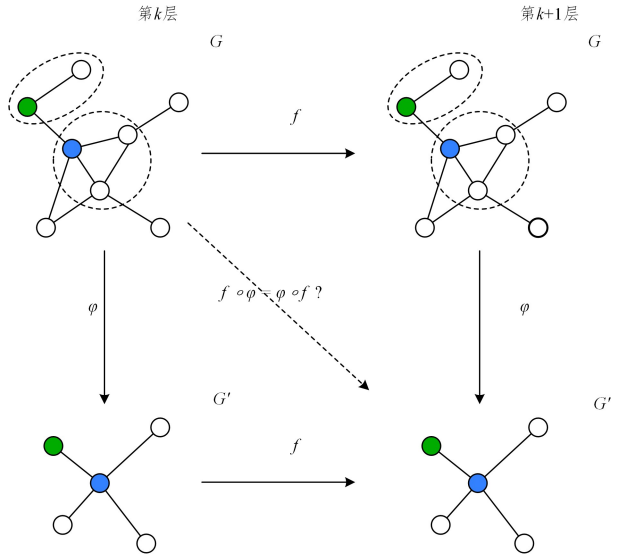

图 1 无偏的粗化训练示意图

Fig. 1 Illustration of unbiased-coarsening training

命题 A 试求粗化映射 φ 和对原图 G 上任意两个节点 p 和节点 q 均有定义的消息传播系数 α_{pq} 以及对新图 G' 上任意两个节点 r 和节点 s 均有定义的消息传播系数 α_{rs}，使得下列约束方程组成立(分别记为约束方程 A.1 和约束方程 A.2)：

$$\begin{cases} \boldsymbol{h}_p^{(k+1)} = \sum\limits_{q\in\mathcal{N}(p)} \alpha_{pq}\boldsymbol{h}_q^{(k)} & (A.1) \\ \boldsymbol{h}_r^{(k+1)} = \sum\limits_{s\in\mathcal{N}(r)} \alpha_{rs}\boldsymbol{h}_s^{(k)} & (A.2) \end{cases}$$

3.2 代数推导

接下来我们给出一个新命题 2。

命题 B 若有下列 3 个条件成立，则原命题 1 成立。

1)对于原图 G 和新图 G' 满足，若节点 v 不在节点 u 的邻域中，则 $\alpha_{uv}=0$；

2)对于给定的节点 p，$q\in G'$ 和任意的节点 $s\in\varphi^{-1}(q)$，均满足 $\sum\limits_{r\in\varphi^{-1}(p)}\alpha_{rs}\equiv C_{pq}$，其中 C_{pq} 表示只与 p 和 q 有关的常数；

3)对于任意的节点 p 和节点 q，满足：

$$\alpha_{pq}=\frac{\varphi^{-1}(q)}{\varphi^{-1}(p)}\sum_{r\in\varphi^{-1}(p)}\alpha_{rs}$$

下面给出原命题 A 在约束条件 B.1 下的充要性推导。

$$\begin{aligned} \boldsymbol{h}_p^{(k+1)} &= \sum_{q\in N(p)}\alpha_{pq}\boldsymbol{h}_q^{(k)} \\ &= \sum_{q\in G'}\alpha_{pq}\boldsymbol{h}_q^{(k)} \Leftrightarrow \frac{1}{\varphi^{-1}(p)}\sum_{r\in\varphi^{-1}(p)}\boldsymbol{h}_r^{(k+1)} \\ &= \sum_{q\in G'}\alpha_{pq}\frac{1}{\varphi^{-1}(q)}\sum_{s\in\varphi^{-1}(p)}\boldsymbol{h}_s^{(k)} \\ &\Leftrightarrow \frac{1}{\varphi^{-1}(p)}\sum_{r\in\varphi^{-1}(p)}\sum_{s\in G}\alpha_{rs}\boldsymbol{h}_s^{(k)} \\ &= \sum_{q\in G'}\alpha_{pq}\frac{1}{\varphi^{-1}(q)}\sum_{s\in\varphi^{-1}(p)}\boldsymbol{h}_s^{(k)} \\ &\Leftrightarrow \sum_{r\in\varphi^{-1}(p)}\sum_{s\in G}\alpha_{rs}\boldsymbol{h}_s^{(k)} \\ &= \sum_{q\in G'}\sum_{s\in\varphi^{-1}(p)}\frac{\varphi^{-1}(p)}{\varphi^{-1}(q)}\alpha_{pq}\boldsymbol{h}_s^{(k)} \\ &\Leftrightarrow \sum_{s\in G}\sum_{r\in\varphi^{-1}(p)}\alpha_{rs}\boldsymbol{h}_s^{(k)} \\ &= \sum_{s\in G}\sum_{q=\varphi(s)}\frac{\varphi^{-1}(p)}{\varphi^{-1}(q)}\alpha_{pq}\boldsymbol{h}_s^{(k)} \\ &= \sum_{s\in G}\frac{\varphi^{-1}(p)}{\varphi^{-1}(\varphi(s))}\alpha_{p\varphi(s)}\boldsymbol{h}_s^{(k)} \\ &\Leftrightarrow \sum_{s\in G}\left(\frac{\varphi^{-1}(p)}{\varphi^{-1}(\varphi(s))}\alpha_{p\varphi(s)}-\sum_{r\in\varphi^{-1}(p)}\alpha_{rs}\right)\boldsymbol{h}_s^{(k)}=0 \end{aligned}$$

为了使得上述公式成立，只需求和号中每一项的系数均为零，即可得到约束方程(记 $\varphi(s)=q$)。

$$\begin{aligned} &\frac{\varphi^{-1}(p)}{\varphi^{-1}(\varphi(s))}\alpha_{p\varphi(s)}-\sum_{r\in\varphi^{-1}(p)}\alpha_{rs}=0 \\ \Leftrightarrow&\alpha_{p\varphi(s)}=\frac{\varphi^{-1}(\varphi(s))}{\varphi^{-1}(p)}\sum_{r\in\varphi^{-1}(p)}\alpha_{rs} \\ \Leftrightarrow&\alpha_{pq}=\frac{\varphi^{-1}(q)}{\varphi^{-1}(p)}\sum_{r\in\varphi^{-1}(p)}\alpha_{rs} \end{aligned}$$

其中，最后一步变换用到了约束条件 B.2，得到的约束公式即为约束条件 B.3，即命题 B 是命题 A 的充分条件。从上述可知，由于充分性，我们将命题 B.3 视作命题 A 的松弛化，并在后续过程中加以分析。

命题 B.3 试求粗化映射 φ 和对原图 G 上任意两个节点 p 和节点 q 均有定义的消息传播系数 α_{pq} 以及对新图 G' 上任意两个节点 r 和节点 s 均有定义的消息传播系数 α_{rs}，使得下列约束方程成立。

$$\alpha_{pq}=\frac{\varphi^{-1}(q)}{\varphi^{-1}(p)}\sum_{r\in\varphi^{-1}(p)}\alpha_{rs}$$

事实上，若不考虑计算复杂性，对于任意合理的粗化映射 φ 和传播系数 α_{pq} 和 α_{rs}，已经可以定义出完整的算法框架，如算法 2 所示。

算法 2 基于图粗化算法的无偏的图神经网络训练

输入：图 G=(V,E)，特征矩阵 $\mathbf{X}$，标签 Y，消息聚合模型 $GNN_G(W)$，损失函数 $\mathscr{L}$

输出：网络的最优参数 $\mathbf{W}^*$

1. 通过粗化映射 φ 把原图 G 粗化成图 G′，由 φ 诱导的粗化矩阵为 $\mathbf{P}$；

2. 计算粗化后对应的特征矩阵 $\mathbf{X}'=\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{X}$ 和标签 $\mathbf{Y}'=\arg\max(\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{Y})$；

3. 在图 G′上做消息聚合 $\mathbf{h}_p^{(k+1)}=\sum\limits_{q\in\mathcal{N}(p)}\left(\frac{\varphi^{-1}(q)}{\varphi^{-1}(p)}\sum\limits_{r\in\varphi^{-1}(p)}\alpha_{rs}\right)\mathbf{h}_q^{(k)}$，用于更新节点特征；

4. 计算损失函数 $L(GNN_{G'}(W)(\mathbf{X}'),\mathbf{Y}')$，得到最优参数 $\mathbf{W}^*$；

5. 返回 $\mathbf{W}^*$ 作为图神经网络 $GNN_G(W)$ 的最优训练参数。

3.3 基于随机游走的权重转移

本文通过直接定义传播系数 α_{rs} 来达成约束条件 B.3，对算法的计算进行简化。考虑基于随机游走的消息聚合，传播系数为 $\alpha_{rs}=\frac{w_{rs}}{\sum\limits_{u\in G}w_{us}}$，带入公式计算得到：

$$\begin{aligned} \alpha_{pq} &= \frac{1}{\varphi^{-1}(p)}\sum_{r\in\varphi^{-1}(p)}\sum_{s\in\varphi^{-1}(p)}\alpha_{rs} \\ &= \frac{\varphi^{-1}(q)}{\varphi^{-1}(p)}\sum_{r\in\varphi^{-1}(p)}\frac{w_{rs}}{\sum\limits_{u\in G}w_{us}} = \frac{\varphi^{-1}(q)}{\varphi^{-1}(p)}\frac{\sum\limits_{r\in\varphi^{-1}(p)}w_{rs}}{\sum\limits_{u\in G}w_{us}} \\ &= \frac{\varphi^{-1}(q)}{\varphi^{-1}(p)}\frac{w_{pq}}{\sum\limits_{v\in G'}w_{vq}} \end{aligned}$$

可以发现，这与随机游走的消息聚合系数的定义在形式上高度近似！

此时，我们可以得到第一个结论：在新图 G' 上做基于随机游走的消息聚合等价于在原图 G 上做基于随机游走的消息聚合。但是，此时公式的消息聚合系数需要进行一步缩放，多乘了一个常数因子。

而此时限制条件 B.3 的物理意义为：每一个节点 s 到节点 p 对应的原像集内所有节点的权重之和，与到所有节点的权重之和(在权重初始化为 1 时，其值为节点 s 的出度)的比值是一个只与节点 p 相关的常数。

得到的算法如算法 3 所示。

算法 3 基于随机游走的消息聚合的粗化训练

输入：图 G=(V,E)，特征矩阵 $\mathbf{X}$，标签 Y，消息聚合模型 $GN N_G(W)$，损失函数 $\mathscr{L}$

输出：网络的最优参数 $\mathbf{W}^*$

1. 通过粗化映射 φ 把原图 G 粗化成图 G′，由 φ 诱导的粗化矩阵为 $\mathbf{P}$；

2. 计算粗化后对应的特征矩阵 $\mathbf{X}'=\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{X}$ 和标签 $\mathbf{Y}'=\arg\max(\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{Y})$；

3. 在图 G′上做基于缩放随机游走系数的消息聚合 $\mathbf{h}_p^{(k+1)}=\sum\limits_{q\in\mathcal{N}(p)}\left(\frac{\varphi^{-1}(q)}{\varphi^{-1}(p)}\frac{\mathbf{w}_{pq}}{\sum\limits_{v\in G'}\mathbf{w}_{vq}}\right)\mathbf{h}_q^{(k)}$，用于更新节点特征；

4. 计算损失函数 $\mathscr{L}(GNN_{G'}(W)(\mathbf{X}'),\mathbf{Y}')$，得到最优参数 $\mathbf{W}^*$；

5. 返回$\mathbf{W}^*$作为图神经网络$\mathrm{GNN}_G(\mathbf{W})$的最优训练参数。

3.4 基于对称传播的权重转移

注意到约束条件B.3不具有对称性约束，3.3节中提出的基于随机游走的消息聚合系数同样是非对称的。

下面考虑引入对称性的推导，即$\alpha_{pq}=\alpha_{qp}$。

$$\begin{cases}\alpha_{pq}=\dfrac{\varphi^{-1}(q)}{\varphi^{-1}(p)}\displaystyle\sum_{r\in\varphi^{-1}(p)}\alpha_{rs}\\ \alpha_{qp}=\dfrac{\varphi^{-1}(p)}{\varphi^{-1}(q)}\displaystyle\sum_{s\in\varphi^{-1}(p)}\alpha_{rs}\end{cases}$$

两边同时对r和s求和，得到$|\varphi^{-1}(p)|=|\varphi^{-1}(q)|$，这是一个极强的约束条件，记为约束条件C，其物理含义为每一个新图中的节点所对应的原像集是等大的。在此基础上，我们可以得到一个新的消息聚合系数的计算式：

$$\alpha_{pq}=\frac{1}{\sqrt{\varphi^{-1}(p)\varphi^{-1}(q)}}\sum_{r\in\varphi^{-1}(p)}\sum_{s\in\varphi^{-1}(p)}\alpha_{rs}$$

此时注意由拉格朗日恒等式可以近似得到：

$$\sqrt{\boldsymbol{w}_{pq}}\approx\frac{1}{\sqrt{\varphi^{-1}(p)||\varphi^{-1}(q)}}\sum_{r\in\varphi^{-1}(p)}\sum_{s\in\varphi^{-1}(p)}\sqrt{\boldsymbol{w}_{rs}}$$

我们只需要令$\alpha_{pq}=\sqrt{w_{pq}}$和$\alpha_{rs}=\sqrt{w_{rs}}$，上述两个公式就在近似意义上达成了一致，所得到的近似算法如算法4所示。

算法4 对称传播的消息聚合的粗化训练

输入：图G=(V,E)，特征矩阵**X**，标签Y，消息聚合模型$\mathrm{GNN}_G(\mathrm{W})$，损失函数$\mathscr{L}$

输出：网络的最优参数$\mathbf{W}^*$

1. 通过粗化映射φ把原图G粗化成图G′，由φ诱导的粗化矩阵为**P**；
2. 计算粗化后对应的特征矩阵$\mathbf{X}'=\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{X}$和标签$\mathbf{Y}'=\arg\max(\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{Y})$；
3. 在图G′上做对称传播的消息聚合$\mathbf{h}_p^{(k+1)}=\sum_{q\in N(p)}\sqrt{w_{pq}}\mathbf{h}_q^{(k)}$，用于更新节点特征；
4. 计算损失函数$\mathscr{L}(\mathrm{GNN}_{G'}(\mathbf{W})(\mathbf{X}'),\mathbf{Y}')$，得到最优参数$\mathbf{W}^*$；
5. 返回$\mathbf{W}^*$作为图神经网络$\mathrm{GNN}_G(\mathbf{W})$的最优训练参数。

4 启发式的图粗化算法

本章将介绍两种启发式的图粗化算法，时间复杂度为$O(|V|+|E|)$（其中，$|V|$表示图的节点个数，$|E|$表示图中边的条数），远远地快于Loukas在文献[9-10]中提出的需要计算矩阵求逆的算法。

我们考虑将原图上本就相邻的节点聚合到一起，具体的实现方法有边聚合和点聚合两种。这是因为粗化的物理意义在直观上就是将原图中的若干个节点聚合在一起，用一个新节点代替。

需要指出的是，本文并未针对粗化算法的理论保证做约束，仅仅从计算速度出发设计了两种算法，其中代表边或点的重要性程度的评估函数均经过对比实验得出，是经验上的选择。不同的粗化函数或者粗化方式会得到不同的新图G'，显然这会影响后续的神经网络参数的迁移学习的效果。在规模化应用时，一种解决思路是根据不同的数据集的特点，针对性地设计不用的粗化方法，当然这失去了普适性。

故下面介绍两种简单的普适的粗化算法，在5.3节中，通过分析实验结果表明其适用性广且效果良好。

4.1 边聚合

边聚合算法对图G中所有的边e_{uv}均计算一个启发式分数$w=\frac{1}{d(u)^2+d(v)^2}$，代表边的重要性程度，然后将重要性程度低的边的两个端点聚合到一起。具体的流程如算法5所示。

算法5 基于边聚合的粗化算法

输入：图G=(V,E)，特征矩阵**X**，标签**Y**，粗化系数α

输出：图G′=(V′,E′)，特征矩阵**X**′，标签**Y**′

1. 对于边e_{uv}计算权重$\mathbf{w}=\frac{1}{d(u)^2+d(v)^2}$；
2. 计算所有边的权重的α分位数q，将权重高于q的边的端点映射到同一个新的图节点，从而得到粗化映射φ；
3. 通过粗化映射φ把原图G粗化成图G′，由φ诱导的粗化矩阵为**P**，计算粗化后对应的特征矩阵$\mathbf{X}'=\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{X}$和标签$\mathbf{Y}'=\arg\max(\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{Y})$。

4.2 点聚合

点聚合算法将图G中节点i的出度$d(i)$视为其重要性程度，然后将重要性程度低的点随机与其一个邻居节点聚合到一起。具体的流程如算法6所示。注意到此时的聚合具有随机性，且当图的节点个数较多时，在实现中会花费较多的时间。

算法6 基于点聚合的粗化算法

输入：图G=(V,E)，特征矩阵**X**，标签**Y**，粗化系数α

输出：图G′=(V′,E′)，特征矩阵**X**′，标签**Y**′

1. 计算所有节点的出度d(i)的1−α分位数q，将所有权重低于q的节点与它的一个邻居随机映射到同一个新的图节点，从而得到粗化映射φ；
2. 通过粗化映射φ把原图G粗化成图G′，由φ诱导的粗化矩阵为**P**，计算粗化后对应的特征矩阵$\mathbf{X}'=\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{X}$和标签$\mathbf{Y}'=\arg\max(\hat{\mathbf{P}}^{\mathrm{T}}\mathbf{Y})$。

5 实验及效果评估

5.1 实验数据和指标

为了评估前述算法的有效性，本文使用文献[12]中提出的3个节点分类数据集Cora，Citeseer和Pubmed进行实验。数据集的具体说明如表2所列。

表2 实验所使用数据集

Table 2 Statistics of node classification datasets

名称	节点数	边数	类别数
Cora	2708	10556	7
Citeseer	3327	9228	6
Pubmed	19717	88651	3

本文采用标准分割（Standard Split），按默认比例将每个数据集划分为训练集、验证集和测试集3部分，其中训练集用于训练模型，验证集用于验证模型训练效果，测试集仅用于评估模型性能。

本文采用精度（accuracy）（单位为%）、算法计算消耗内存（单位为Mb）以及粗化算法总耗时（单位为s）作为评价指标，用于评估模型节点分类的综合性能和粗化算法的运行速度以及计算开销。

5.2 对比模型和方法

为了验证本文算法的有效性，实验设计共选取3个同质

图神经网络模型作为对照组：GCN[1]，Random Walk[13]和Coarsen GCN[7]。对于本文算法，一共选取4组实验组，由聚合算法1(记为RW)、聚合算法2(记为SQRT)与粗化算法3(记为Edge)、粗化算法4(记为Node)两两搭配而成。

本文的实验主语言为Python，通过PyTorch和DGL[14]实现图神经网络模型的开发，利用交叉熵损失函数Cross Entropy[15]计算训练误差和Adam[16]优化器优化模型参数。

为了进行公平的比较，实验设计统一采取深度为2的图神经网络结构，且隐变量的维度保持一致，这是由于粗化算法的主要优化正在于图规模和特征维度上的缩小，若不一致则会导致无法对内存占用空间进行比较。对于使用图粗化算法框架的几种算法，采用统一的粗化系数 $\alpha=0.5$。

由于模型参数的随机初始化和训练时的提前终止(Early Stop)策略会对最终结果造成影响，实验设计采用每个模型在每个数据集上运行20次取计算结果的平均值的方法。

5.3 实验结果和分析

表3列出了不同的图神经网络模型在数据集Cora、Citeseer和Pubmed上的精度、粗化算法总耗时和算法计算消耗内存的情况，其中最佳的数据用粗体标出，可以得出如下结论：

1)在Cora数据集上，本文算法的最佳精度相比GCN仅有0.5%的损失，表现略优于Coarsen GCN，且此时的计算内存消耗降低了44.98%；粗化算法总耗时平均有高达91.04%的降幅，运算速度提升显著。

2)在Citeseer数据集上，本文算法的最佳精度相比GCN有1.2%的提高，表现优于Coarsen GCN，且此时的计算内存消耗降低了39.86%；粗化算法总耗时平均降幅为89.98%，提升显著。

3)在Pubmed数据集上，本文算法的最佳精度相比GCN有0.4%的损失，表现略优于Coarsen GCN，且此时的计算内存消耗降低了48.23%；粗化算法总耗时平均降幅仍高达89.96%，同样提升显著。

4)本文提出的两种消息聚合方法在同一数据集上的表现差距并不大，基本在1%以内，但是不同的粗化算法在同一数据集上的表现存在差异；而且在Pubmed数据集上，基于边聚合的粗化算法所搭配的两种算法模型均精度损失严重。

5)基于点聚合粗化算法所花费时间更长，消耗的时间的增幅平均达到62.84%；内存开销主要由计算涉及的矩阵的大小所决定，因此在粗化算法框架下内存开销均有所降低，其平均值43.97%与粗化系数 $\alpha=0.5$ 相差不大，符合预期。

表3 不同模型在数据集上的实验结果对比

Table 3 Comparison of experiment results of different methods on datasets

模型	Cora			Citeseer			Pubmed		
	精度	内存	时间	精度	内存	时间	精度	内存	时间
GCN[1]	**0.823**	27.48	—	0.711	52.74	—	**0.799**	135.84	—
Random Walk[16]	0.814	28.42	—	0.715	63.69	—	0.792	135.84	—
Coarsen GCN[7]	0.817	**14.31**	4.451	0.696	31.57	6.270	0.791	**67.92**	26.175
Ours(RW,Edge)	0.789	14.40	**0.174**	0.721	31.72	0.459	0.713	71.89	1.291
Ours(SQRT,Edge)	0.791	14.40	0.184	**0.723**	31.72	**0.386**	0.722	71.89	**1.250**
Ours(RW,Node)	0.809	15.09	0.689	0.685	**31.53**	0.849	0.795	70.32	4.021
Ours(SQRT,Node)	0.818	15.12	0.548	0.696	31.64	0.820	0.781	71.19	3.945

总体来说，实验结果表明，精度损失在1%以内，本文算法在粗化速度上得到了高达90%左右的巨幅提升，在计算内存消耗上亦有明显增加，与粗化系数保持大致相同。

通过分析Pubmed数据集上的表现可以发现，基于边聚合的粗化算法是模型表现不佳的原因。这是因为Pubmed数据集的发散程度较高，有超过40%的节点的度为1，当采用边聚合的粗化算法时容易在局部出现把所有邻居全部聚合到一起的不合理情况。实验结果也表明，针对不同的数据集应该采用对应的合理的粗化算法。

另一方面可以发现，基于点聚合的粗化算法花费的时间更长，经分析发现，主要的时间开销集中在算法4的步骤1，其中涉及的随机抽取一个邻居节点是非常耗时的。这是因为DGL中并未有内置的函数调用以优化操作，以及这一步的理论时间花费为 $|N(u)|$，即遍历它的邻域，当邻域太大时耗时较长。相比基于边聚合的粗化算法浪费了更多的时间。这说明两种粗化算法在精度和时间上存在区别，与原始数据集的结构特征有关。

此外，大量的工作证实，Random Walk模型在一般情况下确实不如GCN模型表现好，对称性在网络语义表示中起到了关键作用。这可以作为本文提出的RW算法模型不如Coarsen GCN模型的部分解释。当然表4所列的实验结果表明精度损失不明显，精度互有高低，并不构成本质上的差距及模型间的优劣。

5.4 模型特点分析

目前已有诸多优秀的模型结构，如在第1章中已经介绍过各种各样的方法。为了进行统一的对比，表4详细列出了各类方法之间的结构性特点，主要体现在3个方面：在大规模图上的可扩展性、算法是否需要进行预处理以及在消息聚合框架下的算法主体的计算复杂度。

自然，与Coarsen GCN类似，本文算法的框架是在大图上可扩展的，粗化方法可以和诸如邻居采样、子图采样之类方法混用，以达到在特定数据集上调优的目的。

从本质上来说，在 k 阶邻居诱导的子图上进行消息的聚合本身就是指数级的运算，在不丢失信息的情况下，难以将基于采样思想的算法的时间复杂度控制在线性级。而通过预处理，将原图降采样是一个不错的解决思路，这是因为如果能够将整个图都放入内存进行全批次计算，GCN的时间复杂度是线性级。但是此时预处理的时间

成本成为了时间开销的主体，是不能被忽略的。

表 4 部分常见模型结构的特性对比

Table 4 Comparison of characteristics of some common model structures

模型	可扩展性	预处理	计算复杂度
GraphSAGE[2]	√	—	指数级
Fast GCN[3]	—	—	一阶线性
S-GCN[4]	√	—	指数级
PPRGo[17]	—	√	高阶线性
Cluster GCN[5]	√	√	一阶线性
GraphSAINT[6]	√	—	一阶线性
SGC[11]	—	—	一阶线性
Coarsen GCN[7]	√	√	高阶线性
Ours	√	—	一阶线性

对于图神经网络的训练而言，加速就是在信息损失和预处理时间花费上做权衡(Trade-off)，邻居采样和子图采用的理论基础都是如此，粗化亦然。在第 3 章中提及了两个强约束，即约束条件 B.2 和约束条件 C，在实际实现中是难以满足的。也就是说粗化在通常情况下必然是有损压缩，粗化算法所探求的即为在损失尽可能小的情况下，计算速度提升尽可能大。这也是本文算法的核心动机之一。同时，现在的图神经网络训练大多依赖于 GPU，粗化虽然只是常数级别的缩减，但足以应对大多数导致 OOM(Out of Memory)的情况，对于一些在小显存 GPU 上的训练有明显的帮助。

实际应用的场景下，应该根据不同的情况采取不同的方法，图粗化提出了一种新的解决问题的视角和思路。

结束语 本文提出了一种新的基于图粗化算法的图神经网络训练框架，和现有的其他图神经网络训练方法相比，本文方法是基于图粗化算法的，有本质动机上的区别，并且能够在精度损失不明显的情况下，将模型的训练速度提高 90%左右。该方法改进了已有图粗化算法计算时间开销大的问题，通过提出两种启发式的图粗化算法，并结合新的无偏的消息聚合系数计算公式，得到了简单快速的训练框架，这是基于已有结构的二次创新。

未来的工作包括进一步研究更精妙的图粗化算法，以期能够实现图粗化算法和图神经网络训练上下游相结合的端到端的图神经网络结构，以及对前述提到的两个约束条件进行更细致的分析，给出更严格的理论约束，用于指导算法设计。

参 考 文 献

[1] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[J]. arXiv: 1609.02907, 2016.

[2] HAMILTON W, YING Z, LESKOVEC J. Inductive representation learning on large graphs[J]. Advances in Neural Information Processing Systems, 2017: 30, 1024-1034.

[3] CHEN J, MA T, XIAO C. Fastgcn: fast learning with graph convolutional networks via importance sampling[J]. arXiv: 1801.10247, 2018.

[4] CHEN J, ZHU J, SONG L. Stochastic training of graph convolutional networks with variance reduction[J]. arXiv: 1710.10568, 2017.

[5] CHANG 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 & Data Mining. 2019: 257-266.

[6] ZENG H, ZHOU H, SRIVASTAVA A, et al. Graphsaint: Graph sampling based inductive learning method[J]. arXiv: 1907.04931, 2019.

[7] HUANG Z, ZHANG S, XI C, et al. Scaling up graph neural networks via graph coarsening[C]// Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021: 675-684.

[8] ANDERSEN R, CHUNG F, LANG K. Local graph partitioning using pagerank vectors[C]// 2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06). IEEE, 2006: 475-486.

[9] LOUKAS A. Graph Reduction with Spectral and Cut Guarantees[J]. Journal of Machine Learning Research, 2019, 20(116): 1-42.

[10] LOUKAS A, VANDERGHEYNST P. Spectrally approximating large graphs with smaller graphs[C]// International Conference on Machine Learning. PMLR, 2018: 3237-3246.

[11] WU F, SOUZA A, ZHANG T, et al. Simplifying graph convolutional networks [C] // International Conference on Machine Learning. PMLR, 2019: 6861-6871.

[12] YANG Z, COHEN W, SALAKHUDINOV R. Revisiting semi-supervised learning with graph embeddings[C]// International Conference on Machine Learning. PMLR, 2016: 40-48.

[13] 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.

[14] WANG M, ZHENG D, YE Z, et al. Deep graph library: A graph-centric, highly-performant package for graph neural networks [J]. arXiv: 1909.01315, 2019.

[15] GOODFELLOW I, BENGIO Y, COURVILLE A. Deep learning [M]. The MIT press, 2016.

[16] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. arXiv: 1412.6980, 2014.

[17] BOJCHEVSKI A, KLICPERA J, PEROZZI B, et al. Scaling graph neural networks with approximate pagerank[C]// Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 2464-2473.

CHEN Yufeng, born in 1999, postgraduate. His main research interest is graph neural networks.

HUANG Zengfeng, born in 1986, Ph.D, researcher, doctoral advisor. His main research interests include machine learning algorithms and theory, big data and theoretical computation computer science.

(责任编辑：喻藜)