

融合全局与局部细粒度特征的 图相似度度量算法

车翔玖, 李 良

(吉林大学 计算机科学与技术学院, 长春 130012)

摘 要: 由于计算两个图之间的精确相似度通常属于非确定性多项式时间难解(NP-hard)问题, 因此需要解决精度和速度的权衡问题, 本文提出了一种基于池化的图神经网络方法, 有效地融合了图数据的全局粗粒度交互特征以及子图间节点的细粒度交互特征, 在保证精度的前提下进一步降低了计算量。实验结果表明, 本文方法在真实图数据集上表现出良好的性能, 相较于现有方法, 既提高了准确性, 又提升了计算效率。

关键词: 计算机应用技术; 图神经网络; 图相似度计算; 图池化; 图编辑距离

中图分类号: TP391.4 **文献标志码:** A **文章编号:** 1671-5497(2025)07-2365-07

DOI: 10.13229/j.cnki.jdxbgxb.20231143

Graph similarity measurement algorithm combining global and local fine-grained features

CHE Xiang-jiu, LI Liang

(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Abstract: Since calculating the exact similarity between two graphs is usually an NP-hard problem, the tradeoff between precision and speed needs to be addressed. In this paper, a pooling-based graph neural network method is proposed, which effectively integrates the coarse-grained interaction features of the graph data and the fine-grained interaction features of the nodes between the subgraphs, and further reduce the computational cost while ensuring the accuracy. The experimental results show that the proposed method has good performance on real graph data sets. Compared with existing methods, the proposed method not only improves the accuracy but also improves the computational efficiency.

Key words: computer application technology; graph neural network; graph similarity computation; graph pooling; graph edit distance

0 引 言

近年来, 图神经网络(Graph neural network, GNN)在化学、生物信息学、推荐系统等领域的应

用逐渐增多。研究人员已经开发了多种 GNN 模型^[1,2], 一些模型通过学习节点嵌入实现节点分类任务^[3-6], 另一些模型将节点嵌入汇总为图级嵌入^[7-9]或利用数据增强^[10]实现图分类任务。

收稿日期: 2023-10-23.

基金项目: 国家自然科学基金项目(62172184); 吉林省科技发展计划项目(20200401077GX, ZD21100).

作者简介: 车翔玖(1969-), 男, 教授, 博士. 研究方向: 计算机图形学及大数据可视化. E-mail: chexj@jlu.edu.cn

图分析的一类基本问题是计算图之间的相似度或距离。其中,图编辑距离(Graph edit distance, GED)和最大公共子图(Maximum common subgraph, MCS)是较常用的度量方法,但其精确计算需要较大的计算开销。为解决这一问题,一种简单有效的方法是利用GNN将输入图编码为图级嵌入,随后通过交互计算相似度。但这种方法存在缺陷,即图的差异可能存在于微小子结构中,而图级交互会忽视细粒度的交互特征。

近期,Bai等^[11]提出的SimGNN模型通过两图之间节点级和图级交互实现相似度计算。此外,他们还提出只考虑节点交互的GraphSim模型^[12],对两图的节点嵌入相似度矩阵进行卷积操作^[13]。Ling等^[14]提出的MGMM模型通过节点与图的跨级别交互和图级交互进行相似度计算。Qin等^[15]的EGSC模型通过多层次GNN特征融合网络计算相似度,并利用知识蒸馏在保证准确率的同时加速推理。Ranjan等^[16]的GREED模型分析GED性质,并设计归纳偏置完成相似度计算。Zhuo等^[17]的ERIC模型引入对齐正则化,在训练阶段对GNN编码器施加节点-图约束,推理阶段直接使用学到的图级嵌入计算相似度。然而,在网络模型构建时还需考虑处理以下问题:

(1)计算复杂度较高的问题:随着图规模的增加,节点对数量呈平方级增长,计算所有节点对的相似度需要大量时间和资源。此外,许多节点对的相似度可能非常接近,这意味着存储这些冗余信息会造成资源浪费。

(2)信息丢失的问题:直方图特征通常将连续数据分成离散区间,不可避免地会导致原始数据信息丢失。同时,离散化处理也会导致直方图特征不支持反向传播,使其难以用于训练模型。

为解决上述问题,本文提出一种基于池化的图相似度计算(Pooling-based graph similarity computation, PGSC)方法。该方法从图中筛选关键节点并生成更具代表性的子图,计算子图间节点的相似度,避免冗余的相似度计算,有效降低了计算量。随后将子图相似度矩阵重构并传递至模型预测层,解决了直方图特征信息丢失的问题,使模型能够更有效地进行参数训练。与几种近似GED计算方法和基于GNN的方法相比,本文方法在精度与计算量方面均具有良好的表现。

1 相关工作

1.1 GED

$GED(G_i, G_j)$ 表示图 G_i 和 G_j 之间的编辑距离,定义为将 G_i 转化为 G_j 的最优路径上的编辑操作次数,其中编辑操作包括边的删除或插入、节点的删除或插入以及节点的重新标记。若两图同构,则其GED为0。GED能够捕获两个图之间的结构和特征差异,因此作为图相似度度量方法在图搜索查询中被广泛采用。

图1展示了两个图之间的GED示例,左图和右图的GED为2,需要2个编辑操作:①插入边,②节点重新标记。

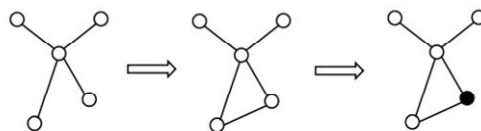


图1 GED示例

Fig. 1 Example of GED

为了更好地拟合端到端回归任务,本文不直接估计两个图之间的GED,而是将其转换为0~1的真实相似度得分来训练模型。首先对GED进行归一化^[18]:

$$nGED(G_i, G_j) = \frac{GED(G_i, G_j)}{(|V_i|, |V_j|)/2} \quad (1)$$

式中: $|V_i|, |V_j|$ 为对应图的节点数。

然后,采用指数函数 $\lambda(x) = e^{-x}$ 将归一化的GED转换为 $(0, 1]$ 的相似度分数,即 $\tilde{y} = e^{-nGED(G_i, G_j)}$ 。

1.2 GCN基本原理

在本文模型中,为了实现两个输入图之间的节点级细粒度交互和图级粗粒度交互,均需要对节点嵌入进行计算。图卷积网络(Graph convolutional networks, GCN)^[3]是目前比较常用的节点嵌入模型,其核心操作是通过聚合节点的邻域信息来生成节点的下一层表示,定义如下:

$$x_i^{l+1} = \sigma_1(\text{AGG}(\{x_j^l, \forall j \in N_i\}, x_i^l)) \quad (2)$$

式中: x_i^l 为节点 i 在第 l 层的特征向量; N_i 为节点 i 的邻居集合; x_j^l 为邻居节点 j 在第 l 层的特征向量; σ_1 为激活函数;AGG($\cdot$)表示邻域信息聚合操作。

2 本文模型

本文提出的 PGSC 方法是一种端到端的神经网络方法,可将一对图数据映射为相似度分数,其结构如图 2 所示。首先,对每个图的节点表示进行更新。其次,通过两图的图级嵌入交互和子图节点级嵌入的相互作用,获取两图在不同粒度的交互特征。最后,将两种粒度的交互特征相结合,传入一个全连接神经网络,得到相似度预测分数。模型各部分的具体操作如下。

2.1 生成节点嵌入

本文采用 GCN 对节点嵌入进行更新,因为 GCN 是一种通用的 GNN 模型,所以可以应用于各种类型的图数据,包括社交网络、推荐系统、生物信息学中的蛋白质互作网络等。这种通用性使其在多个领域都有广泛应用。此外,GCN 既能捕获节点的局部邻居信息,也能考虑全局图的结构,从而在学习节点表示时能够充分考虑节点之间的关系。

孪生网络架构^[19]专门用于学习两个输入之间的相似性,已在视觉识别、视频分割、句子相似度分析等学习任务中取得较好效果。本文使用基于孪生网络架构的多层 GCN,即两个输入图经过一个共享参数的多层 GCN 网络实现节点嵌入更新。

2.2 图级交互

在进行图级交互之前,需要先获得两个输入图的图级嵌入。本文计算图级嵌入的方法借鉴 SimGNN^[11]的注意力机制,因为它能够有效地捕获图的全局信息,而不仅仅依赖于节点级信息。假设通过 GCN 层后输入图 G_i 得到的对应特征矩阵为 $X_i \in \mathbb{R}^{|V_i| \times d}$,其中 d 为每个节点的特征向量维度。

首先计算输入图 G_i 的全局背景向量 $c_i \in \mathbb{R}^d$,

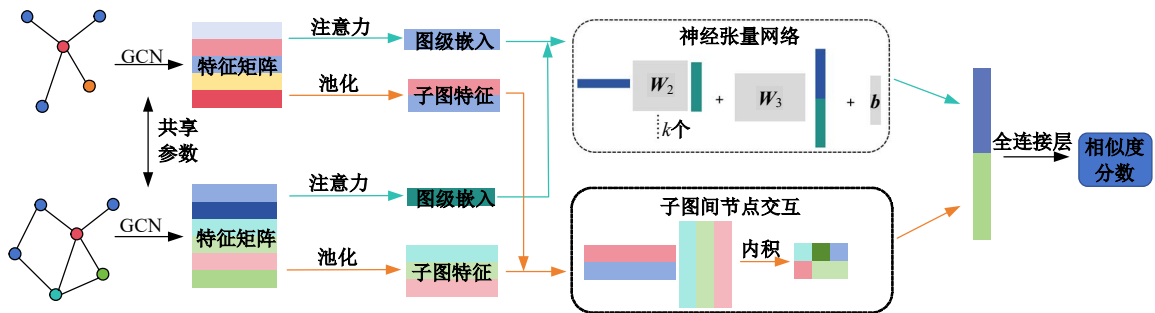


图 2 PGSC 结构

Fig. 2 PGSC structure

如式(2)所示:

$$c_i = \tanh\left(\frac{1}{|V_i|} W_1 \sum_{n=1}^{|V_i|} x_n\right) \quad (3)$$

式中: $W_1 \in \mathbb{R}^{d \times d}$ 为可学习矩阵; $x_n \in \mathbb{R}^d$ 为节点的特征向量。

其次,计算图中每个节点对于图的重要度 $a_n = \sigma_2(c_i^\top x_n)$,其中激活函数将重要度值限制在(0,1)。

最后,利用每个节点的特征向量以及重要度计算得到图的全局表示:

$$H_i = \sum_{n=1}^{|V_i|} a_n x_n \quad (4)$$

如果直接利用两个图级嵌入做内积的方法对两图之间的关系进行建模,往往会导致交互不足或弱交互。而神经张量网络(Neural tensor network, NTN)^[20]在处理复杂数据(如图像、视频和序列等)方面已经展现出较好的性能,具备学习非线性关系的能力,能更好地捕捉数据中的复杂特征和模式。因此,本文利用 NTN 建模两个图级嵌入的交互关系。具体操作方法如式(5)所示:

$$T(H_i, H_j) = \sigma_3(H_i^\top W_2^{[1,k]} H_j + W_3 [H_i; H_j]^\top + b) \quad (5)$$

式中: $W_2^{[1,k]} \in \mathbb{R}^{d \times d \times k}$, $W_3 \in \mathbb{R}^{k \times 2d}$ 均为可学习矩阵; $[\cdot]$ 表示拼接操作; $b \in \mathbb{R}^k$ 为偏置项; k 为超参数,可以控制两个输入图产生的相互作用分数数量。

2.3 节点级交互

正如引言所述,两个图之间的差异常存在于微小子结构中,这些差异难以通过图级嵌入充分反映。然而,计算所有节点对的相互作用既涉及高计算成本又可能导致信息冗余。针对此问题,本文采用图池化策略获取代表性子图,并在这些子图间进行相似度计算。具体操作如下:

$$\begin{cases} \theta = \frac{X_i p}{\|p\|} \\ idx = \text{rank}(\theta, \lambda|V_i|) \\ \tilde{\theta} = \sigma_4(\theta(idx)) \\ \tilde{X}_i = X_i(idx) \odot \tilde{\theta} \end{cases} \quad (6)$$

式中: $p \in \mathbb{R}^d$ 为可学习向量; λ 为池化率, 可以控制生成的子图中包含的节点数; $\text{rank}(\cdot)$ 为进行节点排序操作并返回 θ 中最大的 $\lambda|V_i|$ 个节点的索引; $\odot$ 为对应行做数乘运算; $\tilde{X}_i$ 为生成的子图所对应的特征矩阵。

对生成的子图特征矩阵 $\tilde{X}_i$ 和 $\tilde{X}_j$ 做矩阵乘法, 得到子图相似度矩阵 $s = \sigma_5(\tilde{X}_i \tilde{X}_j^T)$, 并将其重构为相似度向量 S 传入预测层, 以避免直方图特征可能产生的问题。

2.4 相似度分数预测

通过图级交互和节点级交互, 分别获得了两种级别的交互特征: 一种是粗粒度的全局交互特征, 另一种是细粒度的局部交互特征。本文将这两种级别的交互特征结合在一起, 以获得两输入图全面的交互数据。然后, 将它们送入 MLP 中进行处理, 获得最终的相似度得分。具体操作如式(7)所示:

$$y = \sigma_6(\text{MLP}[\text{T}(H_i, H_j); S]) \quad (7)$$

在训练阶段使用均方误差损失函数将其与真实相似度评分进行比较:

$$\ell = \frac{1}{N} \sum_{i=1}^N (\tilde{y} - y)^2 \quad (8)$$

式中: $\tilde{y}$ 为真实相似度值; y 为预测值; N 为一个训练批次中图对的数量。

3 实验结果及分析

3.1 数据集

在 AIDS、LINUX 和 IMDB3 个图数据集上对本文方法进行评估。使用 PyG 提供的标准数据加载器“GEDDataset”来加载这些数据集。

AIDS(即 AIDS700nef)数据集^[11]包含了 700 个用于抗病毒筛选的省略氢原子化合物结构图, 每个图最多包含 10 个节点, 每个节点都被标记为 29 种类型中的一种。

LINUX 数据集^[21]由 LINUX 内核生成的程序依赖图组成, 其中每个图代表一个函数。图中节点表示语句, 边表示不同语句的依赖关系。本文选择 1 000 个图, 每个图的节点数都不超过 10 个。

IMDB 数据集^[22]由 1 500 个电影演员的自我中心网络组成, 若两位演员出现在同一部电影中, 则存在一条边。为了测试本文方法的可扩展性, 使用了完整的 IMDB 数据集。

3.2 评价指标

为全面评估本文模型在相似性计算任务上的性能, 采用以下 5 个指标进行评估:

均方误差(Mean squared error, MSE): 衡量真实相似度值和预测值的平均平方差。MSE 值越小, 表示模型的性能越好。

斯皮尔曼秩相关系数(Spearman's rank correlation coefficient, ρ)和肯德尔秩相关系数(Kendall's rank correlation coefficient, τ): 评估预测结果与真实排名结果的相关性。 ρ 和 τ 值越大, 表示模型的性能越好。

精确率@ k (Precision at k , $p@k$): 通过将预测的前 k 个结果与真实的前 k 个结果的交集除以 k 进行计算。在本文中, 取 $k=10$ 和 $k=20$ 。 $p@k$ 值越大, 表示模型的性能越好。

3.3 训练策略

关于模型的体系结构, 实验设定 GCN 的层数为 3, 并使用 ReLU 作为激活函数。GCN 的各层输出维数分别设置为 64、32 和 16。NTN 层的 k 值设置为 16。在子图节点的交互方面, 将池化比例设置为 0.5。使用两个全连接层来计算最终的预测结果, 其中两个层的维数分别为 16 和 1。对于训练, 将 batch size 设置为 128, 使用 Adam 算法进行优化, 学习率设置为 0.001, 迭代次数设置为 10 000 次。

3.4 实验结果及分析

本文选用的对照方法分为两类, 即快速近似 GED 计算方法和基于神经网络的方法。

第一类包括 3 种经典的 GED 计算方法: A*-Beamsearch (Beam)^[23]是 A* 算法在次指数时间内的一种变体; Hungarian^[24, 25]和 VJ^[26, 27]分别是基于 Hungarian 二部图匹配算法和 Volgenant-Jonker 算法的三次方时间复杂度算法。

第二类包括以下 7 种神经网络架构的模型:

(1) SimpleMean: 通过图中所有节点嵌入的未加权平均值生成图级嵌入。

(2) HierarchicalMean 和 HierarchicalMax^[28]: 基于全局均值、最大池化的 GCN 架构生成图级嵌入。

(3) AttDegree: 使用节点度的自然对数作为

权重计算图级嵌入。

(4) AttGlobalContext 和 AttLearnableGlobal-Context(AttLearnableGC)^[11]:均利用全局背景向量计算图级嵌入。其中,后者对全局背景向量增加了具有可学习权重的非线性转换。

(5) SimGNN^[11]:将提出的 AttLearnableGC 和两组节点嵌入交互中提取的直方图特征相结合进行相似度计算。

其中,前 6 种方法均为 SimGNN 中利用不同生成图级嵌入方法作图相似度计算的对照实验。

对照实验数据均来自 SimGNN,3 个数据集上的实验结果分别如表 1~表 3 所示。本文模型在 3 个数据集上的所有指标均表现出较好的性能,这表明在关键子图之间计算节点级交互,相较于冗余地计算所有节点之间的交互更具意义,并且有效地降低了计算负担。

值得注意的是,本文模型在 IMDB 数据集上的 4 项评价指标均取得良好性能,进一步验证了本文模型的可扩展性。

图 3 对 2 个查询示例进行了可视化。图中的第 1 行和第 3 行分别呈现了 AIDS 和 LINUX 数据集的查询图及其真实相似度排名情况,第 2 行和

表 1 AIDS 数据集结果

Table 1 Results on AIDS data set						
	方法	MSE	ρ	τ	p@10	p@20
快速近似算法	Beam	12.090	0.609	0.463	0.481	0.493
	Hungarian	25.296	0.510	0.378	0.360	0.392
	VJ	29.157	0.517	0.383	0.310	0.345
GNN 方法	SimpleMean	3.115	0.633	0.480	0.269	0.279
	HierarchicalMean	3.046	0.681	0.629	0.246	0.340
	HierarchicalMax	3.396	0.655	0.505	0.222	0.295
	AttDegree	3.338	0.628	0.478	0.209	0.279
	AttGlobalContext	1.472	0.813	0.653	0.376	0.473
	AttLearnableGC	1.340	0.825	0.667	0.400	0.488
	SimGNN	1.189	0.843	0.690	0.421	0.514
	PGSC (pool-hist)	2.266	0.859	0.686	0.442	0.521
	PGSC	2.154	0.861	0.689	0.468	0.550

表 2 LINUX 数据集结果

Table 2 Results on LINUX data set						
	方法	MSE	ρ	τ	p@10	p@20
GNN 方法	快速 Beam	9.268	0.827	0.714	0.973	0.924
	近似 Hungarian	29.805	0.638	0.517	0.913	0.836
	算法 VJ	63.863	0.581	0.450	0.287	0.251
	SimpleMean	16.950	0.020	0.016	0.432	0.465
	HierarchicalMean	6.431	0.430	0.525	0.750	0.618
	HierarchicalMax	6.575	0.879	0.740	0.551	0.575
	AttDegree	8.064	0.742	0.609	0.427	0.460
	AttGlobalContext	3.125	0.904	0.781	0.874	0.864
	AttLearnableGC	2.055	0.916	0.804	0.903	0.887
	SimGNN	1.509	0.939	0.830	0.942	0.933
	PGSC (pool-hist)	0.678	0.978	0.882	0.950	0.943
	PGSC	0.429	0.981	0.887	0.961	0.945

表 3 IMDB 数据集结果

Table 3 Results on IMDB data set						
	方法	MSE	ρ	τ	p@10	p@20
GNN 方法	SimpleMean	3.749	0.774	0.644	0.547	0.588
	HierarchicalMean	5.019	0.456	0.378	0.567	0.553
	HierarchicalMax	6.993	0.455	0.354	0.572	0.570
	AttDegree	2.144	0.828	0.695	0.700	0.695
	AttGlobalContext	3.555	0.684	0.553	0.657	0.656
	AttLearnableGC	1.455	0.835	0.700	0.732	0.742
	SimGNN	1.264	0.878	0.770	0.759	0.777
	PGSC (pool-hist)	0.910	0.899	0.779	0.841	0.831
	PGSC	0.736	0.904	0.784	0.839	0.841

第 4 行展示了本文模型返回的图,并按照预测的相似度分数进行了排序,其中 AIDS 数据集中不同颜色节点代表不同的化学元素。图下方的数字表示相应图对的预测相似度分数,验证了本文模型的良好性能。

3.5 消融实验

为验证本文方法的有效性,以 SimGNN 为基础模型进行消融实验。其中,本文引入 PGSC (pool-hist)作为一种改进方法,将原本计算所有节点交互改为计算子图之间节点交互,并提取其直方图特征。而本文的 PGSC 方法则取消了直方图特征,直接将子图相似度矩阵重构并传入预测

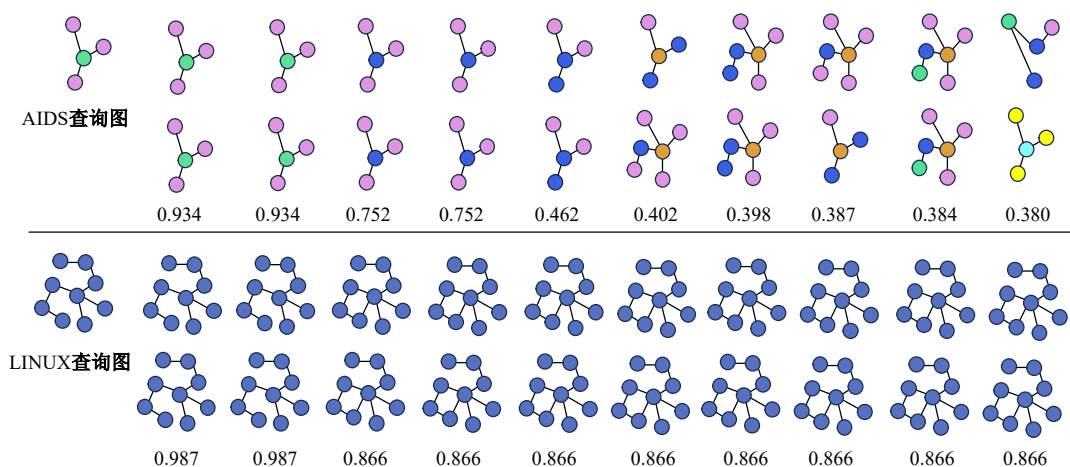


图3 查询示例可视化

Fig. 3 Query examples visualization

层,以最大程度保留交互信息。实验结果如表1~3表所示,充分验证了本文方法的有效性。

4 结束语

本文提出了一种基于池化的图相似度计算模型,能够有效地计算输入图之间的图级交互特征,并且利用图池化技术获得输入图的重要子图,有效地计算子图之间的细粒度交互特征。相较于现有经典算法,本文模型在图相似度计算方面表现出更高的精度、更小的计算量和更强的可扩展性。然而,本文所提模型仍然存在一些局限性:一方面,当图的结构较为复杂,包含大量细粒度的信息时,仅通过子图节点级交互及图级交互可能难以充分捕捉图的全部信息;另一方面,当图呈稀疏结构,节点之间的连接相对较少时,生成子图可能导致出现孤点,从而使节点之间的交互不足。在未来的工作中,将继续探索更加适合的GNN方法和图池化技术,以获得更具代表性的节点嵌入和更重要的子图结构,从而进一步提高模型在图相似度计算任务中的性能,拓宽其应用领域。

参考文献:

- [1] 徐冰冰,岑科廷,黄俊杰,等. 图卷积神经网络综述[J]. 计算机学报, 2020, 43(5): 755-780.
Xu Bing-bing, Cen Ke-ting, Huang Jun-jie, et al. A survey on graph convolutional neural network[J]. Chinese Journal of Computers, 2020, 43(5): 755-780.
- [2] 刘俊奇,涂文轩,祝恩. 图卷积神经网络综述[J]. 计算机工程与科学, 2023, 45(8): 1472-1481.
Liu Jun-qi, Tu Wen-xuan, Zhu En. Survey on graph convolutional neural network[J]. Computer Engineer-

ing & Science, 2023, 45(8): 1472-1481.

- [3] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J/OL]. [2016-02-16]. <https://arxiv.org/html/1609.02907>.
- [4] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs[J]. Advances in Neural Information Processing Systems, 2017, 30: 1024-1034.
- [5] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J/OL]. [2017-06-16]. <https://arxiv.org/html/1710.10903>.
- [6] Brody S, Alon U, Yahav E. How attentive are graph attention networks? [J/OL]. [2021-06-26]. <https://arxiv.org/html/2105.14491>.
- [7] Ying Z, You J, Morris C, et al. Hierarchical graph representation learning with differentiable pooling[J]. Advances in Neural Information Processing Systems, 2018, 31: 4800-4810.
- [8] Gao H Y, Ji S W. Graph U-nets[C]//Proceedings of the 36th International Conference on Machine Learning. New York: Curran Associates, Inc., International Machine Learning Society, 2019: 2083-2092.
- [9] Lee J, Lee I, Kang J. Self-attention graph pooling [C]//Proceedings of the 36th International Conference on Machine Learning. New York: Curran Associates, Inc., International Machine Learning Society, 2019: 3734-3743.
- [10] Luo Y, McThrow M, Au W Y, et al. Automated data augmentations for graph classification[J/OL]. [2022-08-11]. <https://arxiv.org/html/2202.13248>.
- [11] Bai Y, Ding H, Bian S, et al. SimGNN: A neural network approach to fast graph similarity computation [C]//Proceedings of the Twelfth ACM International

- Conference on Web Search and Data Mining. Yew York: ACM, 2019: 384-392.
- [12] Bai Y, Ding H, Gu K, et al. Learning-based efficient graph similarity computation via multi-scale convolutional set matching[J]. AAAI, 2020, 34(4): 3219-3226.
- [13] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2012, 25: 1106-1114.
- [14] Ling X, Wu L, Wang S, et al. Multilevel graph matching networks for deep graph similarity learning [J]. IEEE Transactions on Neural Networks and Learning Systems, 2023, 34: 799-813.
- [15] Qin C, Zhao H, Wang L, et al. Slow learning and fast inference: Efficient graph similarity computation via knowledge distillation[J]. Advances in Neural Information Processing Systems, 2021, 34: 14110-14121.
- [16] Ranjan R, Grover S, Medya S, et al. GREED: a neural framework for learning graph distance functions [J]. Advances in Neural Information Processing Systems, 2022, 35: 22518-22530.
- [17] Zhuo W, Tan G. Efficient graph similarity computation with alignment regularization[J]. Advances in Neural Information Processing Systems, 2022, 35: 30181-30193.
- [18] Qureshi R J, Ramel J Y, Cardot H. Graph based shapes representation and recognition[C] //Graph-Based Representations in Pattern Recognition: 6th IAPR-TC-15 International Workshop. Berlin: Springer, 2007: 49-60.
- [19] Bromley J, Guyon I, LeCun Y, et al. Signature verification using a "siamese" time delay neural network [J]. Advances in Neural Information Processing Systems, 1993, 6: 737-744.
- [20] Socher R, Chen D, Manning C D, et al. Reasoning with neural tensor networks for knowledge base completion[J]. Advances in Neural Information Processing Systems, 2013, 26: 926-934.
- [21] Wang X, Ding X, Tung A K H, et al. An efficient graph indexing method[C] //Proceedings of the 28th IEEE International Conference on Data Engineering, Piscataway, USA, 2012: 210-221.
- [22] Yanardag P, Vishwanathan S V N. Deep graph kernels[C] //Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, USA, 2015: 1365-1374.
- [23] Neuhaus M, Riesen K, Bunke H. Fast suboptimal algorithms for the computation of graph edit distance [C] //Structural, Syntactic, and Statistical Pattern Recognition, Berlin, Germany, 2006: 163-172.
- [24] Kuhn H W. The Hungarian method for the assignment problem[J]. Naval Research Logistics Quarterly, 1955, 2(1/2): 83-97.
- [25] Riesen K, Bunke H. Approximate graph edit distance computation by means of bipartite graph matching[J]. Image and Vision Computing, 2009, 27(7): 950-959.
- [26] Fankhauser S, Riesen K, Bunke H. Speeding up graph edit distance computation through fast bipartite matching[C] //Graph-Based Representations in Pattern Recognition: The 8th IAPR-TC-15 International Workshop, Berlin, Germany, 2011: 102-111.
- [27] Jonker R, Volgenant T. A shortest augmenting path algorithm for dense and sparse linear assignment problems[J]. Computing, 1987, 38(4): 325-340.
- [28] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering[J]. Advances in Neural Information Processing Systems, 2016, 29: 3844-3852.