基于元结构匹配与有偏采样的图相似度计算方法

安丽霞[1)] 吴安彪[1)] 袁 野[2)] 孙思琪[1)] 王国仁[2)]

[1)](东北大学计算机科学与工程学院 沈阳 110167)

[2)](北京理工大学计算机学院 北京 100081)

摘 要 作为图分类、图相似搜索等诸多图数据分析任务的核心步骤,图相似度计算一直是备受研究者们所重视的一个热点问题.由于传统图相似度计算方法的复杂性,无法适用于实时计算节点较多的图相似度任务,针对此问题研究者们提出了新型的基于图神经网络的图相似度计算方法.然而这些算法虽有效加快了图相似度的计算,但是仍然存在两方面的不足,从而影响了他们的性能:(1)多数现有工作通过节点级或图级嵌入的比较来衡量图之间的相似度,忽略了大图中丰富的局部结构特征;(2)所有现有工作均随机采样生成图对数据,导致样本包含的结构不均匀,训练所得模型只对部分特定结构敏感因而误差较大.为此,本文提出了一种新颖的基于元结构匹配与有偏采样的图相似度计算方法 MB-GSC(Meta-Structure Matching and Biased Sampling based Graph Similarity Computation).首先提出 GSE(Graph Structure Extraction)算法提取图中元结构并构建图的结构分布向量,然后基于此向量提出有偏采样策略 RSG(Representative Sample Generation)进行代表性样本的生成,用于后续模型训练.同时,提出算法 MSA(Meta Structure Alignment)对提取到的元结构进行最优匹配对齐,从而获取公有结构形状差异与特有结构数量差异,进而构建蕴含有效的局部相似信息的子结构相似向量.最后,在模型中集成节点级成对比较相似向量、图级神经张量网络相似向量、子结构相似向量进行图对相似性计算.为验证算法的有效性,采用 5 个评估指标在 4 个真实数据集上与基准方法进行了大量对比实验,对模型性能进行全面评估.实验结果验证了本文所提算法 MB-GSC 能够更准确且高效地计算图之间的相似度,在 GED 预测、MCS 预测任务上的准确度比现有模型分别提升 11.16%、7.45%,且在保证相同准确率的同时使训练样本数平均减少 54%.

关键词 图相似度计算;图神经网络;图编辑距离;图嵌入;最大公共子图

中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2023.01513

Graph Similarity Computation Method Based on Meta-Structures Matching and Biased Sampling

AN Li-Xia[1)] WU An-Biao[1)] YUAN Ye[2)] SUN Si-Qi[1)] WANG Guo-Ren[2)]

[1)](*School of Computer Science and Engineering, Northeastern University, Shenyang* 110167)

[2)](*School of Computer, Beijing Institute of Technology, Beijing* 100081)

Abstract As the core step of the data analysis tasks such as graph classification and graph similarity search, graph similarity computation has always been a research focus and has been paid much attention by researchers. Due to the complexity of the traditional graph similarity computation algorithms, it can't be applied to the task of real-time calculation of graph similarity between graphs with many nodes. To solve the problem, researchers have proposed new graph similarity computation methods based on graph neural networks. However, although these algorithms effectively

收稿日期:2022-07-28;在线发布日期:2023-01-09.本课题得到国家自然科学基金(61932004,62225203,U21A20516)资助.**安丽霞**,硕士研究生,中国计算机学会(CCF)学生会员,主要研究方向为图神经网络、图相似度计算.E-mail:anlixia567@163.com.**吴安彪**,博士研究生,中国计算机学会(CCF)学生会员,主要研究方向为图数据库、图神经网络.**袁 野**(通信作者),博士,教授,国家优秀青年基金获得者,中国计算机学会(CCF)高级会员,主要研究领域为云计算、大数据管理(包括图数据管理、不确定数据管理、数据隐私保护)、P2P 计算等.E-mail:yuan-ye@bit.edu.cn.**孙思琪**,硕士研究生,中国计算机学会(CCF)学生会员,主要研究方向为社交网络分析.**王国仁**,博士,教授,长江学者,国家杰出青年科学基金获得者,中国计算机学会(CCF)会员,主要研究领域为不确定数据管理、数据密集型计算、非结构化数据管理、分布式查询处理与优化技术.

speed up the computation of graph similarity, there are still two deficiencies that affect their performance: (1) Most existing works measure the similarity between graphs by comparing node-level or graph-level embeddings, ignoring rich local structural features in large graphs; (2) All existing works randomly sample the graph pairs, resulting in uneven structures contained in the samples, so the model obtained from training is only sensitive to some specific structures, which makes the error relatively large. To address the above two drawbacks, this paper proposes a novel graph similarity computation method MB-GSC (Meta-structure Matching and Biased Sampling based Graph Similarity Computation) based on meta-structure matching and biased sampling. Firstly, the GSE (Graph Structure Extraction) algorithm extracts the meta-structures of the graph and construct the structure distribution vector of the graph. And then, a biased sampling strategy RSG (Representative Sample Generation) is proposed to generate representative samples based on structure distribution vector for subsequent model training. Simultaneously, the algorithm MSA (Meta Structure Alignment) is proposed to perform optimal matching and alignment of the extracted meta structures, so as to obtain the difference in shape of public structures and the number of private structures, and then construct similarity vectors of substructures containing local similarity information. Finally, the node-level pairwise comparison vector, the graph-level neural tensor network similarity vector, and the substructure similarity vector are integrated in the model to calculate the similarity of graph pairs. In order to comprehensively evaluate the effectiveness and performance of the MB-GSC algorithm, plenty of experiments are carried out on 4 real data sets within 5 evaluation indicators. Experimental results verify that the proposed algorithm MB-GSC outperforms other benchmark methods, and can calculate the similarity between graphs efficiently and accurately. Specifically, in terms of GED prediction task, the accuracy can be increased by up to 11.16% compared with the benchmark algorithm and 7.45% on MCS prediction task, respectively. Meanwhile, the number of training samples is reduced by an average of 54% while ensuring the same accuracy.

Keywords graph similarity computation; graph neural network; graph edit distance; graph embedding; maximum common subgraph

1 引 言

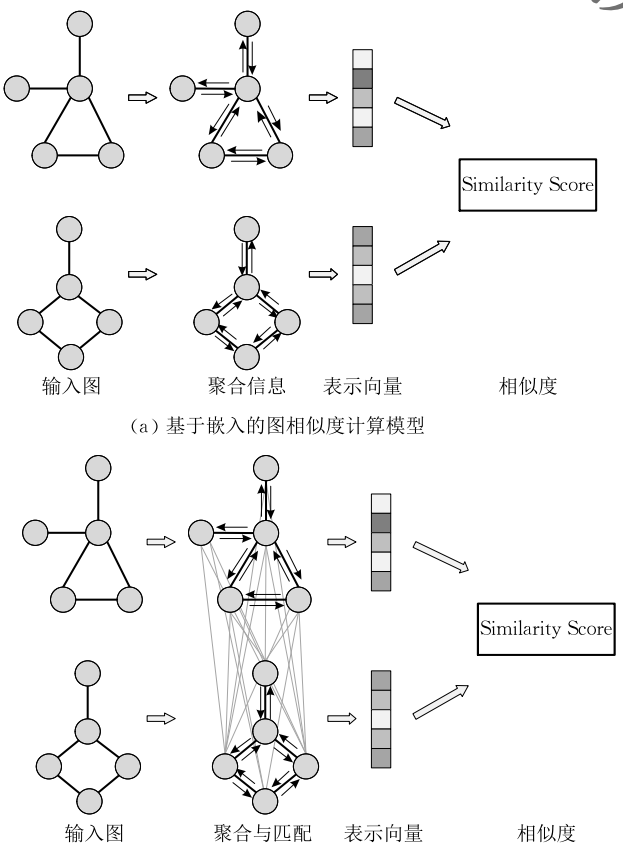
随着互联网与信息技术的飞速发展,生物、化学、交通、社交等领域产生数据的规模呈爆炸式增长,与此同时,数据的结构越来越复杂.图作为一种特殊的数据存储模型,具有以往传统关系数据所不具备的优势,广泛应用于推荐系统、计算机安全、医学图像分析等领域,对大规模的图数据进行有效地分析、处理和挖掘,获取所蕴含的重要信息是非常有意义的研究方向.

图相似度计算是众多图相关的机器学习任务的核心步骤,例如图相似搜索、图分类[1]等,同时也是生命科学、药物研发、模式识别等计算机科学及其交叉领域的基础性问题,在各种现实应用中得到了广泛的关注.例如,在计算机安全中,将程序构建成图进行二进制函数相似性检测[2];在异常检测中,通信图之间的相似性有助于识别网络入侵[3];在神经科学中,分析脑网络图之间的相似性有利于大脑疾病的临床研究[4];在社交网络分析中,探索不同用户的消息图之间的相似性可能会揭示出有意义的行为模式.因此,图相似度计算成为基于图的最重要的应用之一,在理论研究和实际应用方面均具有十分重要的意义.

图编辑距离(Graph Edit Distance,GED)[5]和最大公共子图(Maximum Common Subgraph,MCS)[6]是被广泛使用的衡量两个图之间相似性的度量.然而,计算两个图的精确 GED 与 MCS 是 NP 困难的,在最坏情况下需要指数级时间复杂度,最先进的方法无法在合理的时间内计算超过 16 个节点的图之间的相似度精确值[7].由于图结构数据的复杂性,高效、准确的图相似度计算仍然存在明显挑战.近年来,图神经网络作为一种基于图结构的深度学习方法,已经被广泛地结合在各类传统图分析算法上,能

够以端到端的方式有效地处理图上的复杂任务，为图相似度计算提供了更具前景的解决方案.

近年来，研究人员提出了一些具有代表性的图相似度计算模型，这些模型可以归纳为基于嵌入的图相似度计算模型、基于匹配的图相似度计算模型两类，如图 1 所示. 图 1(a)嵌入整个图为一个图级向量，然后计算向量之间相似度作为图对的相似度，这种方法的局限性在于其没有关注更细粒度的差异信息而预测精度不高. 图 1(b)将每个节点嵌入到一个低维向量，并且成对匹配来计算相似度，通常通过构建成对节点相似度矩阵的直方图特征实现，这种方法的局限性在于将相似矩阵简单划分为直方图可能无法捕获关键交互信息，且节点比较过程具有至少平方时间复杂度，因而消耗了大量的时间. 上述现有的两类图相似度计算模型均没有关注图中重要的局部结构特征，而对于节点较多的图，通常包含丰富的局部结构特征，只关注图级嵌入或只考虑节点级嵌入都不能很好地捕获图中的一些重要局部结构特征，例如，蛋白质分子由许多氨基酸组成，氨基酸由原子组成，而氨基酸决定了蛋白质分子的不同功能，在分析蛋白质分子的功能时，只关注整个蛋白质分子(图级嵌入)或只考虑原子(节点级嵌入)，将不能捕获分子图之间的功能差异. 为了获取子图级差异，现有研究或将大图划分为多个子图，或从超图角度将图粗化并以超边表示子图，目前没有从图的局部结构提取角度出发的相关研究. 此外，现有工作均采用随机方法生成图对，这将有可能导致采样图对中包含的结构不均匀，使得模型只对一部分特定结构敏感而造成误差，同时由于随机生成图对难以控制模型的学习过程，会导致学习效率较低.



(a) 基于嵌入的图相似度计算模型

(b) 基于匹配的图相似度计算模型

图 1　图相似度计算模型

所以针对图神经图相似度计算的研究面临以下挑战：(1) 对于具有层次属性的大图，如何充分讨论图的局部结构特征？(2) 如何选取尽可能少的代表性样本使得训练过程包含均匀图对结构的同时达到效率与准确率的有效提升？(3) 如何集成输入图对之间的多层次交互信息，并端到端地计算图对相似性？(4) 如何在效率与准确性之间实现良好均衡？

为了应对以上挑战，本文提出了一种基于元结构匹配与有偏采样的图相似度计算方法 MB-GSC. 具体来说，首先设计了元结构提取算法并生成图的初始结构分布向量，然后根据初始结构分布向量进行聚类，以识别不同结构的图，进而提出有偏的代表性样本生成策略，目的是对不同图对结构充分均匀采样，并提高学习效率. 其次，针对输入图对，提出了新的针对图对的子结构相似向量生成算法，通过元结构匹配与对齐提取公有结构形状差异与特有结构数量差异，此步骤的目的在于充分比较图之间的局部结构特征. 此外，根据节点属性对图节点排序，目的在于将同构图统一为一种表示形式，进而使相关矩阵对节点排列具有表示不变性，以降低学习难度. 本文主要创新点如下：

(1) 首次从图的子结构提取与分析角度出发，提出了一种新的基于元结构匹配与有偏采样的图相似度计算模型，可以有效提取与聚合局部特征来进行子图级比较，以弥补现有许多模型的表示能力不足或计算成本高的问题；

(2) 设计了图结构提取算法与代表性样本生成策略，能够高效提取图的关键子结构，并解决采样图对不均匀对模型性能的限制；

(3) 提出子图级比较算法，首先通过元结构匹配寻找子结构间的最优对齐，进而通过公有结构与特有结构构造子结构相似向量，最后集成图级、节点级、子图级相似性表示图对差异；

(4) 通过在 4 个真实数据集上执行 GED、MCS

预测任务来进行对比实验，验证了所提出的模型的有效性和性能提升效果.

本文剩余部分首先简单介绍图相似度计算的相关工作(第 2 节)，并给出必要的基本概念及问题定义(第 3 节)；然后介绍了元结构提取及结构分布向量生成算法(第 4 节)；进而，详细介绍了 MB-GSC 图相似度计算框架，以及节点级、图级、子图级比较过程，其中，着重介绍子图级交互，通过元结构匹配与对齐获得公有结构与特有结构，并生成最终的子结构相似向量(第 5 节)；为了说明模型性能，在真实数据集上对算法进行了实验评估与结果分析(第 6 节)；最后，对全文整体工作进行总结(第 7 节).

2 相关工作

本章节主要对与图神经网络、图相似度计算密切相关的工作进行简单介绍与总结.

2.1 图卷积神经网络

图卷积神经网络(Graph Convolutional Neural Network，GCN)将深度神经网络技术推广到非欧氏空间数据：图数据. 借助于卷积神经网络对局部结构的建模能力及图上普遍存在的节点依赖关系，GCN 成为最重要的图上的深度学习模型[8].

现有 GCN 模型大致分为谱方法与空间方法. Bruna 等人[9]在 2013 年提出第一个图卷积神经网络，他们基于图谱理论从卷积定理出发，在谱空间定义图卷积，后来发展为图卷积领域的谱方法；文献[10]利用切比雪夫展开来近似 K-多项式光谱滤波器；GCN[11]进一步将切比雪夫多项式简化为它的一阶近似以进行分层传播. 对于空间方法，图的卷积是通过聚合其空间邻域节点来表示的. GraphSAGE[12]从局部邻域进行采样并聚合表示；GAT[13]引入了自适应聚合节点邻域表示的注意机制. 此外，如何训练更高效的图卷积神经网络也受到广泛关注，研究人员开始试图训练更深层的图卷积神经网络，以增强模型的泛化能力.

2.2 图相似度计算

2.2.1 基于图论的图相似度计算

传统图相似度计算致力于寻找合适的距离度量来反映图之间的相似性，文献[14]在 1983 年首次提出了图编辑距离(Graph Edit Distance，GED)的概念，将输入图转换为目标图所需的编辑操作的最低成本作为图之间距离. 文献[6]在 1998 年提出基于最大公共子图(Maximal Common Subgraph，MCS)的距离度量，并给出了正式证明，与图编辑距离相比，其优点在于不需要定义编辑操作及其成本. 文献[15]证明了在特定的代价函数下，图编辑距离与最大公共子图是等价的，这意味着任何计算图编辑距离的算法都适用于最大公共子图的计算，反之亦然. 然而这些度量的计算被证明是 NP 完全问题[7]. 对于图编辑距离的计算，主要有两类方法：(1)精确计算方法，如 A*-GED[16]通过动态构造搜索树来表示两个图节点集间的所有可能映射关系，进而转化为路径查找问题，遍历搜索树的过程等同于一个编辑路径求解的过程. 随后产生了许多基于改进的 A*算法以求解 GED 问题，文献[17]使用 Hausdorff 编辑距离函数作为启发式函数，在时间复杂度上有所降低；(2)近似计算方法，由于精确 GED 计算复杂度较高，精确算法只适用于小规模图，因此复杂度较低的近似 GED 技术受到更多关注. 此类方法通常通过剪枝策略或启发式来逼近精确值，以快速求解问题，例如文献[18]将 GED 问题转化为基于指派的二部图匹配问题，进而通过匈牙利算法来求解；A*-Beamsearch[19]通过在每一层保留预先设置的束大小的节点继续扩展，剪掉其余较差的点，从而压缩搜索空间.

2.2.2 基于图神经网络的图相似度计算

图相似度学习的目标是学习一个基于图神经网络的能够度量两个图之间相似性的函数. SimGNN[20]首次将图相似学习作为回归任务，利用直方图特征与神经张量网络分别建模节点级与图级交互，GCN 层和注意层由 A*[16]求解的精确 GED 值监督；GraphSIM[21]在 SimGNN 基础上进行扩展，直接匹配两组节点嵌入，以不同层 GCN 为两个图的节点生成向量表示，以捕获复杂的节点级交互；GMN[22]提出不仅在每个图内传播节点表示，而且要在其他图的注意邻居之间传播；HGMN[23]先对图进行层次聚类，得到不同阶段下的多组节点嵌入进行对齐比较差异；MGMN[24]加入一个节点-图跨级交互层，逐一比较节点与另一个图的所有节点；H^2MN[25]从超图的角度学习图的表示，并将每个超边作为一个子图进行匹配；TaGSim[26]对不同类型的编辑操作建模，以节点/边嵌入构建类型感知的图嵌入并通过神经张量网络计算相似性；PSimGNN[27]通过图划分获取子图，并选取 Top-k 进行比较；除此之外，近期的研究试图结合传统 A*算法与图神经网络模型，更高

效地计算精确的 GED；GENN-A*[28]在 SimGNN 基础上进行扩展，提出动态图嵌入网络，搜索树的状态通过动态图嵌入选择启发式；Noah[29]提出 GNN 的扩展图路径网络 GPN，计算估计成本与束大小以优化 A*搜索方向与搜索空间.

2.2.3　其　他

除了上述工作，还有一类基于图核进行相似度计算的研究工作. 图核是一种基于图的子结构度量图相似性的方法，核将整个图结构表示为包含其基本子结构数量的向量，并使用内积将两个向量插入到希尔伯特空间中. Kashima 等人[30]于 2003 年首次提出了针对图的核函数，使用图上的随机游走来定义核. 受此启发，基于图的各种子结构的核被陆续提出，主要有三类：基于游走和路径的图核[31]、基于子树的图核[32]、基于子图的图核[33]. 但是图核方法具有高时间复杂度，且依赖于子结构刚性定义或统计数据，不能够高效处理节点数较多的图.

3　基本概念与问题定义

本章节对必要的基本概念进行介绍，并给出图相似度计算问题的形式化定义. 表 1 对本文所常用的符号表示及其对应含义进行汇总并作简要说明.

表 1　符号表示及含义

符号	含义	符号	含义
δ_{GED}	GED 距离	A	同簇图对集合
δ_{MCS}	MCS 距离	B	异簇图对集合
G	图	GP	代表性图对集合
V	节点集合	W_{12}	公有结构
E	边集合	W_1/W_2	特有结构
e	编辑操作	M	元结构匹配
$c(e)$	e 的编辑成本	$\mathcal{A}$	元结构对齐
ω	元结构类型	H	节点重叠图
Ω	元结构表	$\boldsymbol{U}$	节点嵌入
$\boldsymbol{u}$	结构分布向量	$\boldsymbol{h}$	图嵌入
s	候选结构	$\mathcal{N}(n)$	n 的邻居集合
S	结构列表	$\boldsymbol{b}$	子结构相似向量

定义 1.　图编辑距离，GED. 图 G_1 与 G_2 的编辑距离 GED 是 G_1 和 G_2 间编辑路径的最小成本，如式(1)所示，其中，编辑路径 $P(G_1,G_2)$ 是从图 G_1 转换到图 G_2 的编辑操作序列，每个编辑操作 e_i 都具有一定的编辑成本 $c(e_i)$，编辑路径的成本定义为其编辑操作的成本之和，图上的编辑操作是指节点或边的插入、删除与替换.

$$\delta_{\mathrm{GED}}(G_1,G_2)=\min_{(e_1,\cdots,e_k)\in P(G_1,G_2)}\sum_{i=1}^{k}c(e_i) \tag{1}$$

定义 2.　最大公共子图，MCS. 如果图 $\tilde{G}$ 是图 G_1 和图 G_2 的公共子图且不存在比图 $\tilde{G}$ 节点数更多的公共子图，则图 $\tilde{G}$ 是图 G_1 和图 G_2 的最大公共子图. 基于 MCS 的距离度量定义为式(2)，其中 $|\cdot|$ 表示该图的节点数.

$$\delta_{\mathrm{MCS}}(G_1,G_2)=1-\frac{|mcs(G_1,G_2)|}{\max(|G_1|,|G_2|)} \tag{2}$$

定义 3.　图相似度计算. 给定输入图集合 $\mathcal{G}=\{G_1,G_2,\cdots,G_n\}$，其中 $G_i=(V_i,E_i,X_i)$，$V_i=\{v_1,v_2,\cdots,v_{|V_i|}\}$ 是图 G_i 的节点集，$E_i=\{e_1,e_2,\cdots,e_{|E_i|}\}$ 是图 G_i 的边集，$X_i\in\mathbb{R}^{|V_i|\times D}$ 表示节点特征，D 是节点特征向量的维数. $\mathcal{M}$ 表示一个可学习的相似性函数 $\mathcal{M}:(G_i,G_j)\to R$，以任意两个图 $G_i,G_j\in\mathcal{G}$ 作为输入，输出一个表示它们相似性的分数 $s_{ij}\in R$.

4　结构提取与代表性样本生成

图相似度计算的主要目标是衡量输入图之间的相似性，因而找到这些图之间共享的关键结构模式尤为重要. 本节对原始图数据结构进行预分析，为之后图对之间的比较作铺垫. 首先对元结构进行介绍，然后提出一种结构分布向量生成算法 GSE，用以高效地提取图结构，最后提出算法 RSG 利用该向量有偏地生成代表性样本.

4.1　元结构

直观地，选取的关键结构应当足够简单而广泛存在，且应该是连通性不同、可解释的模式. 基于此分析，本文选取两类五种结构类型组成元结构表：类星型(Star)、类团(Clique)、类二分团(Biclique)、类星型团(Starclique)以及 k-core 子图，以下给出定义.

定义 4.　类星型，Star. 图 $P=(V,E)$ 中存在一个点 $v_i\in V$，对于至少 $p_h\%$ 的节点 $v_j\in V(i\neq j)$，有 $e_{ij}\in E$；同时，对于 $\forall v_j\in V\backslash v_i$，存在至多 $p_s\%$ 的节点 $v_k\in V\backslash v_i$ 使得 $e_{jk}\in E$，则 P 是 Star.

定义 5.　类团，Clique. 如果图 $P=(V,E)$ 满足至少有 $p_h\%$ 的节点对 $v_i,v_j\in V(i\neq j)$，$e_{ij}\in E$，则 P 是 Clique.

定义 6.　类二分团，Biclique. 如果图 $P=(V,E)$ 满足 $\exists L,R$，使 $L\cup R=V$，$L\cap R=\varnothing$，且对于至少 $p_h\%$ 的节点对 $v_i\in L,v_j\in R$，有 $e_{ij}\in E$ 且对于至多 $p_s\%$ 的节点对 $v_m,v_n\in L$ 或 $v_m,v_n\in R$，有 $e_{mn}\in E$，则 P 是 Biclique.

定义 7.　类星型团，Starclique. 如果图 $P=$

(V,E)满足 $\exists L,R$,使 $L\cup R=V, L\cap R=\varnothing$,且对于至少 $p_h\%$的节点 $v_i\in L$,有 $v_j\in R$ 使 $e_{ij}\in E$,且对至多 $p_s\%$的节点对 $v_m,v_n\in R$,有 $e_{mn}\in E$,则 P 是 Starclique.

定义 8. k-core 子图. 如果 $P_k(k\geq 0)$满足所有节点 v 的度数均不少于 k,即 $\forall v\in P_k, \deg_{P_k}(v)\geq k$,且为图 G 满足该式的最大子图,则 P_k 是图 G 的 k-core 子图.

如图 2 所示,Clique 是具有相对均匀连通性的子图,每个节点具有几乎相同的度数;Star 是不均匀连通子图,存在一个中心节点,度数远大于其他所有节点,其他节点之间几乎没有连接;Biclique 结构中,节点可以明显划分为两个集合,两个集合之间紧密连接,而集合内部节点连接稀疏;Starclique 类似 Biclique,但其中一个集合内部节点紧密连接,也就是中心节点是团结构的星型.

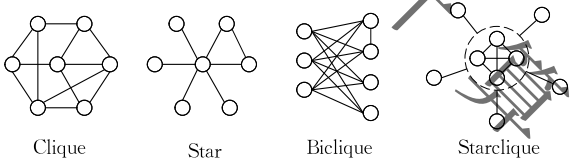


图 2 元结构

上述四种结构对度数与连通性的要求各不相同,但均属于对图的子结构的不精确约束,比如严格的 k-clique 定义要求图的 k 个节点中两两之间均有边相连,而本文定义的 Clique 仅要求大部分节点对之间有边连接,且具体数量由 p_h 动态控制. 为了准确地描述与分析真实世界的图,在上述结构基础上补充一种严格约束,即 k-core 子图,如图 3 所示,图 G 的 k-core 子图 G_k 中每个节点的度数至少为 k.

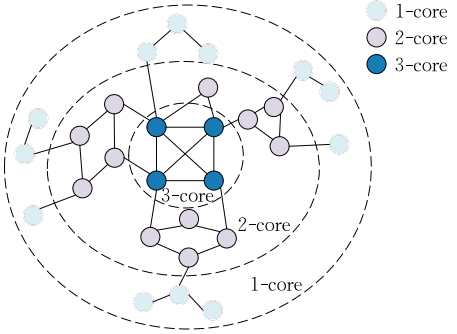


图 3 k-core 子图

4.2 结构分布向量生成

上述五种关键结构共同构成了元结构表,以下将基于此元结构表,在严格约束与不精确约束下,生成图数据的结构分布向量.

图的结构分布向量 $\boldsymbol{u}\in\mathbb{R}^{|\Omega|}$ 定义为一个 $|\Omega|$ 维向量,$|\Omega|$ 表示使用的元结构类型数,其第 i 个分量 $\boldsymbol{u}_i$ 表示元结构 ω_i 所占比例,即 $\boldsymbol{u}_i=n_{\omega_i}\Big/\sum_{i=1}^{|\Omega|}n_{\omega_i}$,$n_{\omega_i}$ 为元结构 ω_i 的数量. 结构分布向量生成过程见算法 1. 首先,将图分解为一组直径不超过 3 的连通分量(第 2 行),具体过程为迭代地选择 C 中当前最大连通分量中度数最高的节点 v,与其邻居节点组成分量 $c\in C$,然后删除与节点 v 连接的所有边,这一步是为了保证生成的分量不重复,直到不能形成更多的分量为止. 以每个 C 中的分量为候选生成元结构表中的各个结构类型(第 3~8 行). 如果相同结构类型的候选结构有大部分节点重叠,则将其合并,并将合并后的结构加入结构列表 S(第 7 行),然后根据图的结构列表 S 计算得到每种结构类型的比例,进而得到该图的结构分布向量 $\boldsymbol{u}$(第 9 行).

算法 1. Graph Structure Extraction(GSE).

输入:图 G,元结构表 Ω

输出:结构分布向量 $\boldsymbol{u}$

1. $S\leftarrow[]$ //结构列表
2. $C\leftarrow decomposition(G)$
3. FOR each type $\omega\in\Omega$
4. FOR each component $c\in C$
5. $S_{\text{candidates}}\leftarrow MetaStructureGeneration(\omega,c)$ //算法 2
6. END FOR
7. $S\leftarrow merge(\omega,S_{\text{candidates}})$
8. END FOR
9. $\boldsymbol{u}=structure\_distribution(S)$
10. RETURN $\boldsymbol{u}$

算法 2 对文献[34]中的算法思想进行了扩展,其描述了分量生成元结构的过程,输入为连通分量 c、结构类型 ω,输出为生成的候选结构 s 与对应节点集合 V_S.

(1) 对于 Clique,首先从当前节点集 V_c 中找到最大团,并将结点加入初始 V_S 集合(第 3 行),对剩余节点降序排序(第 4 行),不断选取具有最大度数的节点,判断该节点是否与集合 V_S 中超过半数的节点连通,若是则将其加入到集合 V_S(第 5~9 行).

(2) 对于 Star,首先选取 V_c 中度数最大节点为中心(第 12 行),剩余节点组成集合 V_S,在中心节点的邻居节点中删除与 V_S 连接紧密的节点,剩余节点与中心节点共同组成 Star 结构(第 13~19 行).

(3) 对于 Biclique,初始化 R 为 V_c 的最大独立集中度数最高的前 m_R 个节点(第 22 行);对剩余与 R

连接超过一半的节点构成的集合，同样取最大独立集中度数最高的前 m_L 个节点组成 L（第 23～29 行）；如果生成的 L、R 集合足够小，则表示不存在 Biclique 结构，直接退出（第 30 行）；对于 V_c 中除 L、R 以外的节点，检查是否有与 R 连接紧密而与 L 连接稀疏的节点，或者与 L 连接紧密而与 R 连接稀疏的节点，如果有则分别加入 R、L（第 31～40 行）.

（4）对于 Starclique，初始化 L 为 V_c 最大团（第 44 行），剩余节点中满足与 L 连接紧密的节点集取最大独立集构成 R（第 45～51 行），对于 V_c 中除 L、R 以外的节点，检查是否存在节点与 L、R 均连接紧密，将其加入 L，是否存在节点与 L 连接紧密而与 R 连接稀疏的节点，将其加入 R（第 52～61 行）.

（5）对于 k-core 子图，本文直接使用其严格定义因而此处不给出 k-core 结构生成算法.

算法 2. Meta Structure Generation（MSG）.

输入：结构类型 ω，连通分量 c

输出：生成结构 s，对应节点集 V_s

1. $V_c \leftarrow$[*node* in c]，$V_s \leftarrow$[] //候选结构节点集
2. IF ω is Clique
3. $V_s \leftarrow maximumClique(V_c)$
4. Sort$V = V_c \backslash V_s$ by node degree（descending）
5. FOR each $u \in V$
6. IF $connect(u, V_s) \geq 50\%$ //$connect(u, V)$表示节点 u 与节点集 V 中节点的连接比例
7. Append node u to V_s
8. END IF
9. END FOR
10. END IF
11. IF ω is Star
12. $v = highestDegreeNode(V_c)$ //选取中心节点
13. $V_s = V_c \backslash \{v\}$
14. FOR each node $u \in \mathcal{N}(v)$
15. IF $connect(u, V_s) > 5\%$
16. remove u from $\mathcal{N}(v)$
17. END IF
18. END FOR
19. $V_s = \mathcal{N}(v) \cup \{v\}$
20. END IF
21. IF ω is Biclique
22. $R \leftarrow MIS(V_c, m_R)$ //MIS 中度数最高的 m_R 个
23. $T \leftarrow$[]
24. FOR each $u \in V_c$
25. IF $connect(u, R) \geq 50\%$
26. Append node u to T
27. END IF
28. END FOR
29. $L' = (V_c \backslash R) \cap T$，$L \leftarrow MIS(L', m_L)$
30. IF $|L| < 5$ or $|R| < 5$ THEN break
31. WHILE true
32. FOR each node $u \in V_c \backslash (L \cup R)$
33. IF $connect(u, L) < 5\%$ and $connect(u, R) \geq 50\%$
34. add u to R
35. END IF
36. IF $connect(u, L) \geq 50\%$ and $connect(u, R) < 5\%$
37. add u to L
38. END IF
39. END FOR
40. END WHILE
41. $V_s = L \cup R$
42. END IF
43. IF ω is Starclique
44. $L \leftarrow maximumClique(V_c)$
45. $T \leftarrow$[]
46. FOR each $u \in V_c$
47. IF $connect(u, L) \geq 50\%$
48. Append node u to T
49. END IF
50. END FOR
51. $R' = (V_c \backslash L) \cap T$，$R \leftarrow MIS(R')$
52. WHILE true
53. FOR each node $u \in V_c \backslash (L \cup R)$
54. IF $connect(u, L) \geq 50\%$ and $connect(u, R) \geq 50\%$
55. add u to L
56. END IF
57. IF $connect(u, L) \geq 50\%$ and $connect(u, R) < 5\%$
58. add u to R
59. END IF
60. END FOR
61. END WHILE
62. $V_s = L \cup R$
63. END IF
64. RETURN s，V_s

4.3　代表性样本生成

现有图相似度计算模型均采用随机方法生成训练图对，而对于图对本身结构相似与否欠缺考虑，这将有可能导致采样不均匀而使模型的学习能力不高，模型稳健性不佳，同时，随机生成图对也使得模型训练效率大为降低. 为此，本文提出一种代表性样本生成策略更有针对性地获取训练图对，旨在提高模型的训练效率与准确率.

代表性样本生成的过程如图 4 所示，在图上执行 GSE 算法提取元结构并生成可解释的结构分布向量，其反映了图中各类子结构的分布情况，基于此向量将图聚类为 k 簇，而后调用算法 RSG 分别从同簇与异簇中选取图组成图对 gp_i，作为代表性样本用于后续训练.

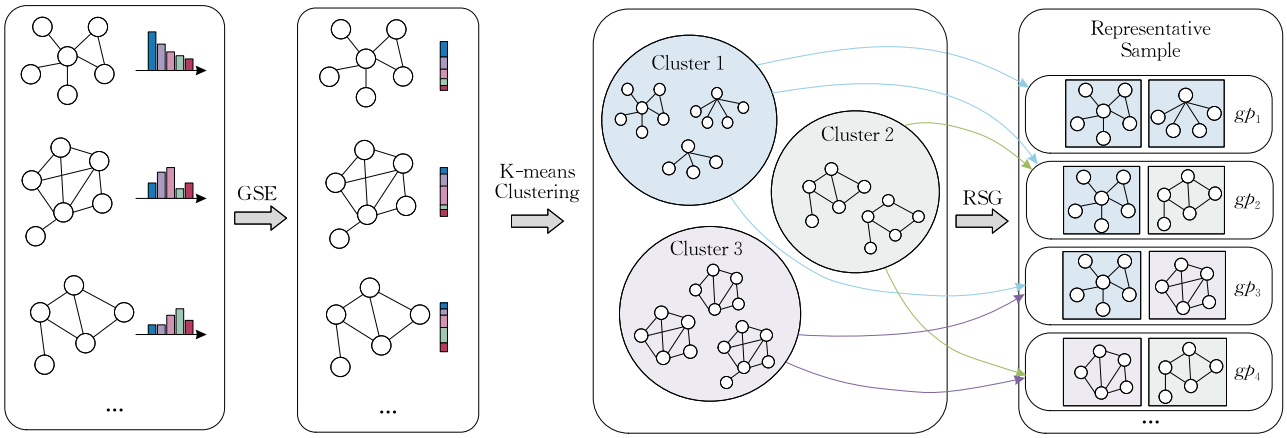


图 4 结构提取与代表性样本生成

由于执行效率高且实现简单，本文选用 K-means 算法进行聚类，算法 3 描述了图对生成的具体过程. 算法第 2～5 行调用算法 GSE 生成结构分布向量集合 E，算法第 6 行将图数据聚为 k 个簇，簇集合为 $C=\{C_1,C_2,\cdots,C_k\}$，其中 $C_i=\{G_1,\cdots,G_{k_i}\},i\in[1,k]$，算法 7～12 行分别组成同簇图对 A 与异簇图对 B，第 13 行对这两类图根据特定比例取样，目的在于以较少量的数据获得更好的性能，提升模型学习效率.

算法 3. Representative Sample Generation (RSG).

输入：图集合 $\mathcal{G}=\{G_1,\cdots,G_n\}$

输出：图对集合 $GP=\{\cdots,(G_i,G_j),\cdots\}$

1. $GP\leftarrow[],E\leftarrow[]$ //结构分布向量集合
2. FOR each graph $G_i\in\mathcal{G}$
3. $\boldsymbol{u}=GraphStructureExtraction(G_i)$
4. add $\boldsymbol{u}$ to E
5. END FOR
6. $C=K\text{-means}Clustering(k,E)$
7. FOR each cluster $C_i\in C$
8. $A=C_i\times C_i$
9. FOR each cluster $C_j\in C$ and $j\neq i$
10. $B=C_i\times C_j$
11. END FOR
12. END FOR
13. $GP\leftarrow\alpha A+\beta B$ //按比例取样
14. RETURN GP

5 图相似度计算框架

本节重点介绍所提出的 MB-GSC 图相似度计算模型，5.1 节给出整体框架概述，5.2、5.3、5.4 节分别描述节点级、图级、子图级交互过程，其中 5.4 节提出元结构匹配与对齐算法并生成子图相似性向量，最后 5.5 节集成多层级相似信息进行最终的相似分数计算.

5.1 框架概述

本文提出的 MB-GSC 图相似度计算框架如图 5 所示，整体包含图级、子图级、节点级三个层次的相似度分数计算，主要流程为结构提取、代表性样本生成、多级嵌入生成、多层级交互、相似向量融合与相

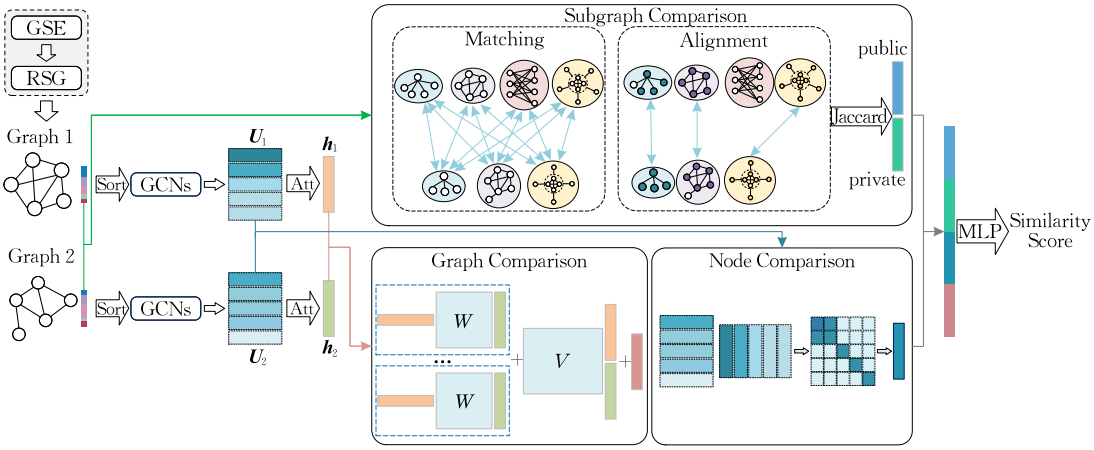


图 5 MB-GSC 框架

似度计算.其中,结构提取与代表性样本生成(灰色区域)已在第 4 节进行讨论.对于输入图对,首先进行排序并使用多层 GCN 生成单一形式的节点嵌入,进而采用注意力机制基于节点嵌入生成图级嵌入,至此,多级嵌入生成过程结束.然后进行各层级比较,图级比较主要使用神经张量网络计算,节点级比较采用成对匹配策略生成相关矩阵并向量化,子图级比较首先对图对的元结构进行匹配,而后寻找其间最优对齐并获得公有结构与特有结构,进而构建子图级相似向量.最后,融合上述 3 级相似信息并应用全连接网络计算最终图相似度分数.

5.2　节点级交互

5.2.1　节点嵌入生成

在现有多种图表示学习方法中,本文选择具有分层传播规则的图卷积网络(GCN[11])以邻居聚合的方式学习初始节点嵌入,考虑了节点所有邻居以及其自身所包含的特征信息,在局部加权平均后生成节点表示.对于图 $G=(V,E,\boldsymbol{X})$,V 为节点集,E 为边集,$\boldsymbol{X}^{N\times K}$ 表示节点特征,其中 N 为节点数,K 为特征数,一层 GCN 按如下方式更新嵌入:

$$\boldsymbol{H}^{l+1}=\sigma(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{l}\boldsymbol{W}^{l}) \tag{3}$$

其中,$\sigma(\cdot)$是激活函数,$\tilde{\boldsymbol{A}}=\boldsymbol{A}+\boldsymbol{I}_N$是带自环的邻接矩阵,$\tilde{\boldsymbol{D}}$ 是定义于增广邻接矩阵$\tilde{\boldsymbol{A}}$ 上的度矩阵,$\tilde{\boldsymbol{D}}_{ii}=\sum_j\tilde{\boldsymbol{A}}_{ij}$,$\boldsymbol{W}^{l}\in\mathbb{R}^{K\times K'}$是第$l$层可学习的权重矩阵,$\boldsymbol{H}^{l}\in\mathbb{R}^{N\times K}$、$\boldsymbol{H}^{l+1}\in\mathbb{R}^{N\times K'}$分别是第 l 层、第 $l+1$ 层的激活矩阵,$\boldsymbol{H}^{0}=\boldsymbol{X}$,初始嵌入用独热编码表示.从单个节点的角度来看,节点特征更新公式如式(4),其中$\mathcal{N}(n)$表示节点 n 的一阶邻居集合(包括节点 n 本身),d_n等于节点 n 的度数加 1,$\boldsymbol{W}\in\mathbb{R}^{K\times K'}$是可学习的权重矩阵.式(4)所表示的卷积操作聚合了来自节点 n 的一阶邻居的特征.

$$\boldsymbol{h}_n^{l+1}=\sigma\Big(\sum_{m\in\mathcal{N}(n)}\frac{1}{\sqrt{d_m d_n}}\boldsymbol{h}_m^{l}\boldsymbol{W}^{l}\Big) \tag{4}$$

通过多层 GCN 嵌套将可以获得充分包含节点邻居信息的节点表示,本文使用 3 层 GCN 获得节点表示.

5.2.2　节点级比较

类似于文献[21,22],为了捕获图之间细粒度的差异,本文通过节点嵌入成对比较计算节点级相似度.假设输入图对 G_i、G_j 的节点嵌入分别为 $\boldsymbol{U}_i$、$\boldsymbol{U}_j$,节点数分别为 N_i、N_j,成对匹配过程如式 $\boldsymbol{S}=\boldsymbol{U}_i\boldsymbol{U}_j^{\mathrm{T}}$,计算得到相关矩阵 $\boldsymbol{S}^{N_i\times N_j}$.对于两个不同大小的图,如上过程必须要求其维度相同,故对于较小的图,插入对应值为 0 的伪节点.因此,节点比较作内积的结果是 N 阶方阵 $\boldsymbol{S}^{N\times N}$,其中 $N=\max(N_i,N_j)$.为了有效地利用节点对相似性,本文将相似性矩阵向量化为 $H(\boldsymbol{S})$.

尽管通过独热编码得到的初始节点表示是基于节点类型的,所以节点排列顺序对最终的相似度分数没有影响,然而在上述节点级比较过程中,由于节点没有统一的顺序,即使是同构图,图 G_i、G_j 的节点嵌入 $\boldsymbol{U}_i$、$\boldsymbol{U}_j$将对应不同的节点编号,这导致了不同的相关矩阵 $\boldsymbol{S}$,而本质上它表示的是同一个图对.如图 6 所示,Graph2 有两种不同的表示形式,进而与 Graph1 成对比较生成了不同形式的相关矩阵.所以未规范排序的节点表示将可能导致相关矩阵的较大差异,进而使得训练过程模型学习难度增大.为此,需要根据一定规则(例如节点中心性)对节点进行排序.

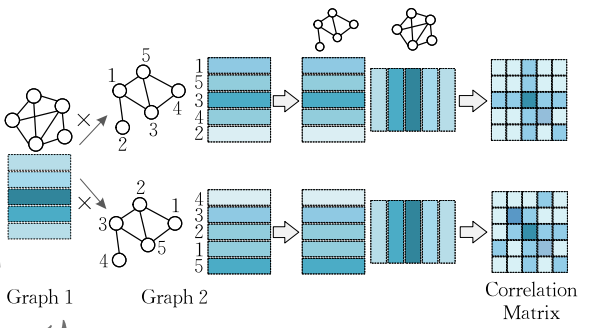


图 6　同构图生成的不同相关矩阵

本文对初始节点表示根据节点 PageRank 值进行排序,式(5)计算得到了各个节点 v 的 PageRank 值 $\boldsymbol{PR}_v$:

$$\boldsymbol{PR}_v=d\cdot\boldsymbol{A}\cdot\boldsymbol{PR}+\frac{1-d}{N}\boldsymbol{1} \tag{5}$$

其中,d 为阻尼因子,$\boldsymbol{A}$ 为邻接矩阵,$\boldsymbol{PR}$ 为节点 PageRank 值向量,N 为节点个数,$\boldsymbol{1}\in\mathbb{R}^N$为 1 向量.

图 7 展示了经过节点排序步骤后的效果,节点嵌入统一形式之后,进而得到的相关矩阵也随之确定,更利于后续的学习任务.

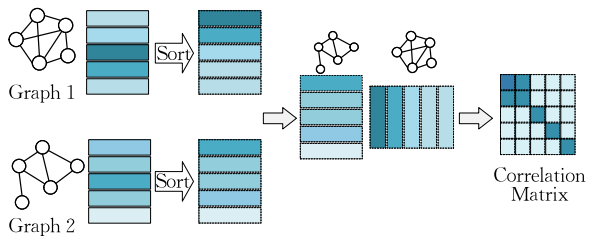


图 7　节点排序后的相关矩阵

5.3　图级交互

5.3.1　图级嵌入生成

为了生成图嵌入,需要有效地组合节点表示.最

简单的方法是对节点嵌入取平均,但是这样无法区分与突出具有不同重要性的节点,为此,本文采用一种注意力机制,对节点给予不同的权值,使具有更大结构意义的节点比其他节点对整体图嵌入的影响更大.

如 5.2.1 节所述,生成的节点嵌入可以表示为 $\boldsymbol{u}\in\mathbb{R}^{N\times D}$,其第 n 行 $\boldsymbol{u}_n\in\mathbb{R}^D$ 表示节点 n 的嵌入,则整个图可以表示为 $\boldsymbol{h}\in\mathbb{R}^D$,如式(6)表示:

$$\boldsymbol{h}=\sum_{n=1}^{N}a_n\boldsymbol{u}_n \tag{6}$$

为了计算每个节点 n 的注意权重 a_n,首先需要计算一个全图上下文向量 $\boldsymbol{c}\in\mathbb{R}^D$,先对所有节点嵌入 $\boldsymbol{u}$ 取平均,然后通过一个非线性的 tanh 激活函数进行转换,这个全图上下文向量 $\boldsymbol{c}$ 的计算如式(7):

$$\boldsymbol{c}=\tanh\left(\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{u}_n\right)\boldsymbol{W}\right) \tag{7}$$

$$a_n=\sigma(\boldsymbol{u}_n^{\mathrm{T}}\cdot\boldsymbol{c}) \tag{8}$$

其中,$\boldsymbol{W}\in\mathbb{R}^{D\times D}$ 为权重矩阵.通过学习权重矩阵 $\boldsymbol{W}$,上下文向量 $\boldsymbol{c}$ 提供了对整个图的结构属性的简单总结.根据上下文向量 $\boldsymbol{c}$,计算节点嵌入与之作内积的结果,得到每个节点 n 的注意权重 a_n,如式(8)所示.其含义在于衡量节点 n 是否对表达整个图特征具有重要意义,可以转化为衡量节点与上下文向量 $\boldsymbol{c}$ 的相似程度,越相似的节点应该获得更高的注意权重.

综合上述内容,整体图嵌入的计算如式(9):

$$\boldsymbol{h}=\sum_{n=1}^{N}\sigma\left(\boldsymbol{u}_n^{\mathrm{T}}\cdot\tanh\left(\left(\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{u}_n\right)\boldsymbol{W}\right)\right)\boldsymbol{u}_n \tag{9}$$

5.3.2 图级比较

对于给定的两个图嵌入 $\boldsymbol{h}_i\in\mathbb{R}^D$,$\boldsymbol{h}_j\in\mathbb{R}^D$,关键问题是如何有效地比较以获取图级相似性.为此,本文使用了文献[35]中提出的神经张量网络(NTN),相比传统的线性层方法,NTN 能更有效地比较两个多维向量.本文使用式(10)计算两个嵌入之间的关系:

$$\mathrm{NTN}(\boldsymbol{h}_i,\boldsymbol{h}_j)=\sigma\left(\boldsymbol{h}_i^{\mathrm{T}}\cdot\boldsymbol{W}^{[1:K]}\cdot\boldsymbol{h}_j+\boldsymbol{V}\begin{bmatrix}\boldsymbol{h}_i\\\boldsymbol{h}_j\end{bmatrix}+\boldsymbol{b}\right) \tag{10}$$

其中,$\boldsymbol{W}^{[1:K]}\in\mathbb{R}^{D\times D\times K}$ 为权重张量,[]表示连接操作,$\boldsymbol{V}\in\mathbb{R}^{K\times 2D}$ 是一个权重向量,$\boldsymbol{b}\in\mathbb{R}^K$ 是偏置向量,$\sigma(\cdot)$ 是激活函数,K 用来控制两个图嵌入比较产生的相似向量的维数的超参数.

5.4 子图级交互

对于具有较多节点的图,通常包含丰富的局部结构,两个图之间的差异主要在于局部子结构,只关注图级嵌入或只考虑节点级嵌入均不能很好地捕获图中的重要局部结构特征.本文首次从图结构分析角度出发,通过提取并比较图的关键子结构生成子图级相似性,以期更好地反映两个大图之间的局部相似性.整个过程包含以下两个步骤:(1)结构匹配与对齐,针对图对的元结构列表进行匹配并对齐,对于完成匹配的公有结构,计算其 Jaccard 相似性,而对于未完成匹配的特有结构,计算其数量差异;(2)子结构相似向量生成,根据前面生成的公有结构与特有结构生成包含形状差异与数量差异的子结构相似向量.

5.4.1 结构匹配与对齐

如 4.2 节所述,根据算法 1 可得到图的结构列表与结构分布向量.针对图对,给定两个结构列表,关键问题是在图的子结构之间寻找最优对齐.

算法 4 描述了元结构对齐过程,输入为两个图的结构列表及对齐关系,输出为图对的公有结构以及各自的特有结构.第 2 行首先计算了二部匹配 $M\subseteq S_1\times S_2$,根据结构类型将 S_1 与 S_2 中的结构进行配对,要求成对的结构属于相同的元结构类型 $\omega\in\Omega$,具体过程由算法 5 描述.对于每个结构对 $(s_1,s_2)\in M$,计算它的公共结构 s,将其添加到 W_{12}(第 3~6 行).最后,将 S_1 与 S_2 的未配对结构分别添加到 W_1、W_2(第 7~11 行).

算法 4. Meta Structure Alignment(MSA).

输入:图 G_1 的结构列表 S_1,图 G_2 的结构列表 S_2,对齐 $\mathcal{A}$

输出:公有结构 W_{12},图 G_1 特有结构 W_1,图 G_2 特有结构 W_2

1. $W_{12},W_1,W_2\leftarrow[]$
2. $M\leftarrow GreedyStructureMatching(S_1,S_2,\mathcal{A})$ //算法 5
3. FOR all structures pair $(s_1,s_2)\in M$
4. $\quad s=CommonStructure(s_1,s_2)$
5. $\quad$ Add s to W_{12}
6. END FOR
7. FOR $i\in\{1,2\}$
8. $\quad$ FOR all structure $s\in S_i\backslash\{s\in S_i\mid\exists p\in M:s\in p\}$
9. $\quad\quad$ Add s to W_i
10. $\quad$ END FOR
11. END FOR
12. RETURN W_{12},W_1,W_2

本文提出了用于匹配的启发式算法,算法 5 描述了详细过程.除了两个图的结构列表,算法还接收一个结构对齐作为输入,此结构对齐可以预先指定部分节点的匹配关系,且可以为空.(1)如果没有给定结构对齐,首先构造节点重叠图 H_i,节点重叠图

的节点集是 S_i，也就是说其节点是 S_i 中的结构，边的权值为结构的节点集之间的 Jaccard 相似性(第 3 行)；然后通过节点重叠图建立 $H_1\times H_2$ 的一个变体图 G，其节点是类型一致的结构对 (s_1,s_2)，边的权值是 H_1 与 H_2 中边权值的乘积(第 4～6 行)；对于图 G，贪婪地迭代选择图中权重最大的边，并删除与这些边不兼容的所有节点，即去除最优之外的结构对(第 7～13 行)；最后，剩余结构中同类型的根据大小降序排列并匹配在一起(第 14～22 行)；(2) 如果给定了结构对齐(可能是部分)，贪婪地迭代选择在对齐 $\mathcal{A}$ 下具有最大 Jaccard 相似性的结构对，并加入最终结果集 M(第 24～33 行).

算法 5.　Greedy Structure Matching (GSM).

输入：图 G_1 的结构列表 S_1，图 G_2 的结构列表 S_2，结构对齐 $\mathcal{A}$

输出：结构匹配 $M\subseteq S_1\times S_2$

1. $M\leftarrow\varnothing$
2. IF $\mathcal{A}=\varnothing$
3. $H_i\leftarrow(S_i,F_i,w_i)$ for $i\in\{1,2\}$, $w_i((s,t))=$ Jaccard(s,t)
4. $V\leftarrow\{(s_1,s_2)\in S_1\times S_2\mid type(s_1)=type(s_2)\}$
5. $E\leftarrow\{((s_1,s_2),(t_1,t_2))\mid(s_1,t_1)\in F_1,(s_2,t_2)\in F_2\}$
6. $G\leftarrow(V,E,w)$, $w(((s_1,s_2),(t_1,t_2)))=\prod\limits_{i\in\{1,2\}}w_i((s_i,t_i))$
7. WHILE $E\neq\varnothing$
8. $(u,v)\leftarrow\arg\max\limits_{(u,v)\in E}w((u,v))$
9. Add u,v to M
10. $X\leftarrow\{x\in V\backslash M\mid(x\cap u\neq 0)\vee(x\cap v\neq 0)\}$
11. $E\leftarrow E\backslash\{(u,v)\}$
12. $G\leftarrow G[V\backslash X]$
13. END WHILE
14. $\overline{S}_i\leftarrow S_i\backslash\{s\in S_i\mid\exists p\in M:s\in p\}$ for $i\in\{1,2\}$
15. FOR all structure $s_1\in\overline{S}_1$
16. FOR all structure $s_2\in\overline{S}_2$
17. IF $type(s_1)=type(s_2)$
18. Add (s_1,s_2) to M
19. $\overline{S}_i\leftarrow\overline{S}_i\backslash\{s_i\}$ for $i\in\{1,2\}$
20. END IF
21. END FOR
22. END FOR
23. END IF
24. ELSE
25. $\overline{S}_i\leftarrow S_i$ for $i\in\{1,2\}$
26. WHILE true
27. $U\leftarrow\{(s_1,s_2)\in\overline{S}_1\times\overline{S}_2\mid type(s_1)=type(s_2)\}$
28. IF $U=\varnothing$ THEN break
29. $(s_1,s_2)\leftarrow\arg\max\limits_{(s_1,s_2)\in U}Jaccard_{\mathcal{A}}(s_1,s_2)$
30. Add (s_1,s_2) to M
31. $\overline{S}_i\leftarrow\overline{S}_i\backslash\{s_i\}$ for $i\in\{1,2\}$
32. END WHILE
33. END ELSE
34. RETURN M

算法针对不同结构类型计算 Jaccard 相似度：对于 Clique，直接计算节点集 Jaccard 值，如式(11)；而对于 Star、Biclique 与 Starclique，节点集被分为 V_1、V_2 两部分，其中 $V=V_1\bigcup V_2$ 且 $V_1\bigcap V_2=\varnothing$，需要分别计算两类节点的 Jaccard 系数并取平均，如式(12). 在 Biclique 与 Starclique 中，V_1、V_2 分别表示左右节点集；Star 中，V_1、V_2 分别为中心节点与其他节点.

$$Jaccard_{\mathcal{A}}(s_1,s_2)=\frac{|\mathcal{A}(V(s_1))\bigcap V(s_2)|}{|\mathcal{A}(V(s_1))\bigcup V(s_2)|}\tag{11}$$

$$Jaccard_{\mathcal{A}}(s_1,s_2)=\frac{1}{2}\sum_{i=1}^{2}\frac{|\mathcal{A}(V_i(s_1))\bigcap V_i(s_2)|}{|\mathcal{A}(V_i(s_1))\bigcup V_i(s_2)|}\tag{12}$$

5.4.2　子结构相似向量生成

如前所述，5.4.1 节为两个图的结构列表寻得最优匹配，通过对齐得到公有结构与各自特有结构，本节利用生成的公有结构与特有结构为图对生成子结构相似向量，补充子图级相似信息.

在算法 5 中两个结构被匹配成对作为图对的公有结构，同时可以根据式(11)、(12)计算得到公有结构对应的 Jaccard 相似度分数，由此可以获得表示结构相似性的一组分数组成的向量. 为了统一嵌入维数便于后续相似分数计算阶段的处理，对所有相似分数根据结构类型取平均值，如式(13)：

$$b_\omega=\frac{1}{k}\sum_{k=1}^{num_\omega}score,\ \omega\in\Omega\tag{13}$$

由此生成的子结构相似向量为 $\boldsymbol{b}\in\mathbb{R}^{|\Omega|}$，其主要反映了图对子结构形状差异，而图中子结构的数量也是构成图对差异性的一部分. 存在一种情况，两个图中某一类型子结构的数量差异较大，则能够匹配成对的子结构将取决于较少的那个图，然而这两个图差别的关键其实在于未匹配的结构. 为此，还需将特有结构数量差异考虑进来，同样地生成特有结构中各结构的比例分布向量. 所以最终的子结构相似向量为 $\boldsymbol{b}\in\mathbb{R}^{2|\Omega|}$，包含公有结构形状差异与特有结构数量差异两部分.

5.5　相似分数计算

前述 5.2、5.3、5.4 节分别获得了图对的节点级、图级、子图级相似向量，本节应用全连接网络逐步降低相似向量的维度，计算最终的预测相似分数

$\hat{s}_{ij}$，使用以下均方误差损失函数将其与地面真实值进行比较：

$$\mathcal{L}=\frac{1}{|\mathcal{T}|}\sum_{(i,j)\in\mathcal{T}}(\hat{s}_{ij}-s_{ij})^2 \tag{14}$$

其中$\mathcal{T}$是训练图对集合，$|\mathcal{T}|$为训练图对的数量，s_{ij}为图对相似度地面真实值.

5.6 时间复杂度分析

MB-GSC 的时间复杂度分析主要包括三部分：(1) 嵌入生成阶段. 该部分时间复杂度为 $O(|E|)$，$|E|$为图的边数；(2) 相似分数计算阶段. 图级相似分数计算的时间复杂度为 $O(D^2K)$，D 为图嵌入维度，K 为神经张量网络的特征映射维度；节点级相似分数计算的时间复杂度为 $O(DN^2)$，$N=\max\{|V_{G_1}|,|V_{G_2}|\}$为较大节点数；子图级相似分数计算中，子结构匹配过程因涉及集合笛卡尔积操作，时间复杂度为 $O(mn)$，m，n 分别为图 G_1，G_2 的子结构数，对齐过程复杂性主要来自结构间的 Jaccard 相似度计算，在最坏情况下为 $O(\tilde{n}^2)$，$\tilde{n}=\max\{|s_1|,|s_2|\}$为两个结构节点数的较大值.

6 实验与分析

本节将通过多组实验验证评估本文算法的有效性，首先对实验数据集、评价指标、对比算法和参数设置进行描述，在此基础上，进行对比实验、消融实验、超参实验，并对实验结果进行分析.

6.1 数据与实验设置

(1) 数据集

本文实验主要使用 4 个来自真实世界不同领域的图数据集：AIDS、IMDB、LINUX、PTC，它们已经被广泛应用于评估图的相似度计算[23,25,32]. AIDS 数据集包含了来自 NCI/NIH7 开发治疗计划中抗病毒筛选化合物集合的 42 687 个化学化合物图，图中的每个节点都与 29 个标签中的一个相关联；LINUX 数据集包含由 Linux 内核生成的 48 747 个程序依赖图(PDG)，在每个 PDG 中，节点是一条语句，边是这两个语句之间的依赖关系，且节点无标签；IMDB-MULTI 包含 1500 个电影演员的自我网络，每个节点代表一名演员，边表示两个演员之间的合作关系；PTC 数据集由 344 张化学化合物网络组成，每个节点具有 19 个标签中的一个. 本文实验使用的上述数据集的基本信息如表 2 所示. 其中，每个数据集的图对为根据给定划分比例得到的训练测试集随机生成的图对，即由 80% 的图与全部图配对组成. 图 8、图 9 可视化了以上数据集图对的 GED、MCS 分布情况.

表 2 数据集基本信息

数据集	图的含义	图数	节点数范围	平均节点数	划分比例	图对数
AIDS	化学化合物	700	[2,10]	8.9	6:2:2	392K
LINUX	程序依赖图	1000	[4,10]	7.6	6:2:2	800K
IMDB-MULTI	自我中心网络	1500	[7,89]	13.0	6:2:2	1800K
PTC	化学化合物	400	[16,103]	30.2	6:2:2	94.6K

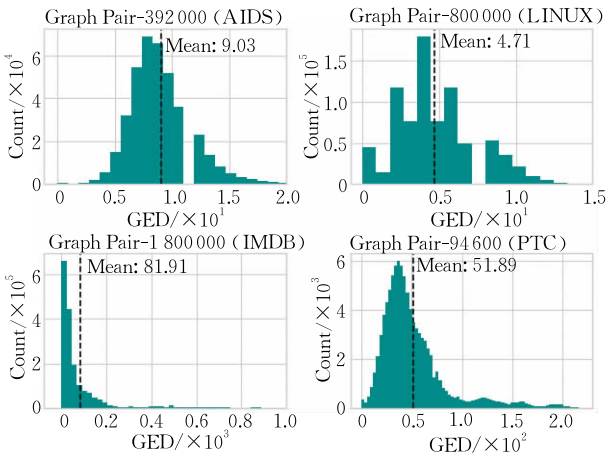


图 8 GED 分布

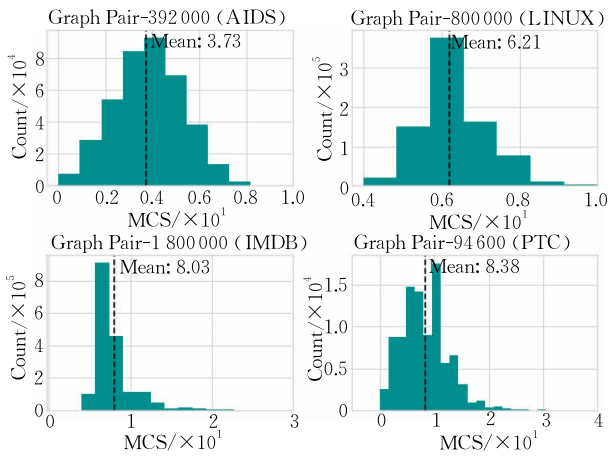


图 9 MCS 分布

(2) 对比算法

由于传统基于图论的图相似度计算方法(如 A*)在性能方面均低于基于 GNN 的算法，本文仅选取现有主流的基于 GNN 的图相似度计算方法进行对

比.在综合大量相关工作的基础上,选择现有 3 类算法与本文方法进行性能对比.

①图级嵌入方法. GCNMean[10]、GCNMax[10]等,直接嵌入整个图为向量并计算嵌入向量之间的相似度,此类算法仅关注粗粒度的图级比较信息.

②节点级匹配方法.以 SIMGNN[20]、GMN[22]、GraphSIM[21]、MGMN[24]为代表,此类算法在图级嵌入比较信息基础之上补充了节点级比较信息或节点-图跨级比较信息.

③子图级比较方法.以 H^2MN[25]、HGMN[23]、PSimGNN[27]为代表,此类算法通过图划分、层次聚类等方式进行子图级比较.

本文在各类算法中选取了具有代表性的方法进行对比,大致介绍如下:

(a) GCNMean[10]、GCNMax[10]. 使用 GCN 学习图的嵌入,平均或最大池化后应用全连接网络计算图的相似度.

(b) SIMGNN[20]. 结合了神经张量网络捕获的图级相似度与从节点嵌入提取的直方图相似度.

(c) GMN[22]. 引入交叉图注意层,通过合并另一个图的注意邻域信息来改进节点嵌入.

(d) GraphSIM[21]. 使用多组节点嵌入生成的相似度矩阵并应用 CNN 进行处理.

(e) MGMN[24]. 设计了节点-图交互机制,获取节点与图之间的跨级交互信息.

(f) H^2MN[25]. 从超图角度学习图的表示,将每个超边作为子图并进行匹配以获取子图相似性.

(3)评估指标

与现有工作一样,本文使用以下 5 个评估指标:运行时间(RT)、均方误差(MSE)、Spearman 等级相关系数(ρ)、Kendall 等级相关系数(τ)、Top-k 准确率($p@k$).其中,RT 评估的是模型的效率,MSE 评估的是图相似度预测的准确性,ρ、τ、$p@k$ 评估的是排名性能,其值越高意味着性能越好.上述指标的具体介绍如下:

①运行时间(RT). 运行时间指模型计算一对图的相似度所用时间.

②均方误差(MSE). 指模型计算的相似度与真实值的平均平方差.

③ Spearman 等级相关系数(ρ). 衡量两个变量依赖性的评价指标,本实验用于评价预测的排序结果与真实排序结果的匹配程度,如式(15)所示.

$$\rho=1-\frac{6\sum(x_i-y_i)^2}{n(n^2-1)} \tag{15}$$

④ Kendall 等级相关系数(τ). 衡量两个变量相关性的统计值,本实验用于评价两个变量的统计依赖性,其取值范围为[−1,1],如式(16)所示,其中 C 为同序对个数,D 为异序对个数.

$$\tau=\frac{C-D}{C+D} \tag{16}$$

⑤ Top-k 准确率($p@k$). 衡量由预测相似度得到前 k 个最近邻的准确率,如式(17)所示,其中 TP_k、FP_k 分别为预测正确 Top-k 集合与预测错误 Top-k 集合.本实验取 $k=10$ 评价模型预测准确度.

$$p@k=\frac{TP_k}{TP_k+FP_k} \tag{17}$$

(4)实验环境与参数设置

实验环境. Linux 操作系统:Ubuntu18.04;Intel Xeon Silver 4214 CPU@2.20 GHz,128 GB RAM,4 个显存为 11 GB 的 NVIDIA RTX 2080Ti GPU;编程环境为 Python 3.6,PyTorch1.1.0[36].

参数设置. 本文实验主要遵循现有工作的实验设置,详细介绍如下.对于每个数据集,以比例 6∶2∶2 随机划分数据集分别作为训练集、验证集、测试集,其他设置如下:

①本文方法.聚类过程中,经过前期实验,令簇内平方和(Inertia)最小寻找到最优聚类个数,因此将 AIDS、LINUX、IMDB、PTC 数据集聚类簇数 k 分别设置为 2、2、3、2,详见 6.4 节超参实验;图对生成过程中,同类图采样比例 $\alpha=70\%$,异类图采样比例 $\beta=80\%$;对于初始节点表示,为具有节点类型的 AIDS、PTC 采用 One-Hot 编码方案,而 IMDB 与 LINUX 的节点无标签,故采用常量编码方案;嵌入生成过程中,GCN 层数设置为 3,使用 ReLU 作为激活函数,GCN 输入尺寸为节点标签数,故 AIDS、LINUX、IMDB、PTC 的输入维数分别为 29、1、1、19,GCN 第 1、2、3 层输出维数分别为 64、32、16;图级交互过程中,NTN 层设置 $K=16$;节点级交互过程中,直方图箱数 $bins=16$;相似分数计算过程中,全连接网络层数为 5,输入输出维数分别为 64 到 32、32 到 16、16 到 8、8 到 4、4 到 1;训练过程中,批量大小 $batch\_size=128$,使用 Adam[37]优化参数,学习率 $lr=5e\text{-}3$,迭代次数 $epoch=1000$,其中后 100 次迭代进行验证,并根据最小的验证损失选择最佳模型.

②对比方法.所有这些方法输入图对为随机组合生成且使用全部图对数据.除此之外,对于 SimGNN,隐藏层个数为 16,NTN 层张量大小 $K=$

16,直方图箱数 $bins=16$,批量大小 $batch\_size=128$,学习率 $lr=$1e-3,丢弃率 $dropout=0.3$;对于 GraphSIM,批量大小 $batch\_size=128$,学习率 $lr=$1e-3,丢弃率 $dropout=0.3$;对于 MGMN,GCN 输出维数均为 100,使用 BiLSTM 作为聚合函数,透视图数 $d=100$,批量大小 $batch\_size=128$,学习率 $lr=$5e-3,丢弃率 $dropout=0.1$;对于 H^2MN,随机种子 $seed=2020$,超边池化率为 $pool\_rate=\{1.0,1.0,0.8\}$,隐藏层个数 100,批量大小 $batch\_size=128$,学习率 $lr=$5e-3,丢弃率 $dropout=0.1$,权重衰减 $weight_decay=$5e-4;对于 GMN,由于复现后效果相比原文较差,此处重用论文中的结果数据.

6.2 与现有方法的对比实验

为了检验本文方法的整体性能,将其与现有 3 类 7 个具有代表性的方法进行比较,分别从准确性、时间空间效率角度进行详细地对比评估.

(1)有效性评估

为了充分对图相似度计算模型的预测准确度进行评估,分别进行了针对图相似性度量 GED、MCS 的预测任务,实验结果如表 3、表 4 所示.实验结果表明:① 本文方法在准确度方面超过现有多数基于图神经网络的图相似度计算方法,图 10、图 11 分别将 GED、MCS 预测错误率可视化(为了使对比更直观,图 10 中超过 10 的按最大值处理),结合该图可知,在 4 个数据集上本文方法 GED 预测性能改进最高百分比分别为{4.26,2.78,10.39,11.16},MCS 预测性能改进最高百分比分别为{0.94,−3.29,7.45,4.05}(相对于最佳基线方法);② 不管是 GED 预测任务还是 MCS 预测任务,对于较大的图数据集 IMDB 与 PTC,本文方法性能提升都优于小图数据集 AIDS、LINUX,且 GED 预测任务在数据集 PTC 上性能提升最明显,为 11.16%,而 MCS 预测

表 3 GED 预测实验结果

数据集	评估指标	GCNMean	GCNMax	SimGNN	GraphSIM	GMN	MGMN	H^2MN	本文	提升/%
AIDS	$MSE(10^{-3})$	2.214	3.423	1.430	1.880	4.310	3.214	2.211	**1.369**	4.26
	ρ	0.653	0.628	0.817	0.851	0.697	**0.869**	0.834	0.840	−3.33
	τ	0.629	0.505	0.667	0.703	0.508	0.702	0.651	**0.780**	9.87
	$p@10$	0.194	0.290	0.419	0.418	0.213	0.383	0.390	**0.426**	1.67
LINUX	$MSE(10^{-3})$	7.541	6.341	2.360	1.076	2.676	5.259	1.561	**1.046**	2.78
	ρ	0.579	0.724	0.943	0.972	0.802	0.915	0.566	**0.981**	0.92
	τ	0.525	0.740	0.822	**0.931**	0.738	0.764	0.675	0.920	−1.18
	$p@10$	0.141	0.541	0.775	0.869	0.862	0.648	0.849	**0.890**	2.41
IMDB	$MSE(10^{-3})$	68.823	58.425	2.964	1.924	3.210	3.145	2.232	**1.724**	10.39
	ρ	0.402	0.449	0.781	0.825	0.725	0.531	0.691	**0.931**	12.84
	τ	0.378	0.354	0.770	**0.821**	0.782	0.467	0.501	0.802	−2.31
	$p@10$	0.219	0.437	0.724	0.813	0.751	0.521	0.498	**0.821**	0.98
PTC	$MSE(10^{-3})$	7.428	8.329	1.873	1.889	1.854	3.650	2.295	**1.647**	11.16
	ρ	0.546	0.506	0.726	0.714	0.670	0.647	0.725	**0.808**	11.29
	τ	0.490	0.468	0.678	**0.719**	0.592	0.708	0.716	0.694	−3.47
	$p@10$	0.210	0.241	0.475	0.541	0.374	0.554	0.499	**0.599**	8.12

表 4 MCS 预测实验结果

数据集	评估指标	GCNMean	GCNMax	SimGNN	GraphSIM	GMN	MGMN	H^2MN	本文	提升/%
AIDS	$MSE(10^{-3})$	6.234	4.156	3.433	2.402	2.234	3.719	2.832	**2.213**	0.94
	ρ	0.756	0.801	0.822	0.858	0.901	0.816	0.807	**0.903**	0.22
	τ	0.498	0.574	0.680	0.798	0.803	0.641	0.639	**0.814**	1.36
	$p@10$	0.347	0.315	0.374	0.505	0.513	**0.757**	0.477	0.525	−30.64
LINUX	$MSE(10^{-3})$	2.689	2.170	**0.729**	3.164	0.794	0.739	1.541	0.753	−3.29
	ρ	0.521	0.714	0.859	0.962	0.939	0.637	0.762	**0.970**	0.83
	τ	0.747	0.784	0.889	0.946	0.934	0.478	0.711	**0.952**	0.63
	$p@10$	0.421	0.459	0.850	0.951	0.949	0.570	0.637	**0.960**	0.94
IMDB	$MSE(10^{-3})$	10.457	10.124	2.451	1.287	0.590	5.128	3.176	**0.546**	7.45
	ρ	0.746	0.841	0.930	0.976	0.941	0.752	0.739	**0.984**	4.56
	τ	0.611	0.619	0.879	0.946	0.920	0.788	0.922	**0.954**	0.84
	$p@10$	0.387	0.451	0.812	0.882	0.875	0.695	0.603	**0.889**	0.79
PTC	$MSE(10^{-3})$	12.441	13.845	5.419	3.975	3.142	2.765	4.807	**2.653**	4.05
	ρ	0.578	0.662	0.712	0.779	0.782	**0.889**	0.682	0.831	−6.52
	τ	0.650	0.688	0.746	0.800	0.792	0.715	0.727	**0.812**	1.50
	$p@10$	0.384	0.402	0.356	0.498	0.584	**0.770**	0.705	0.689	−10.51

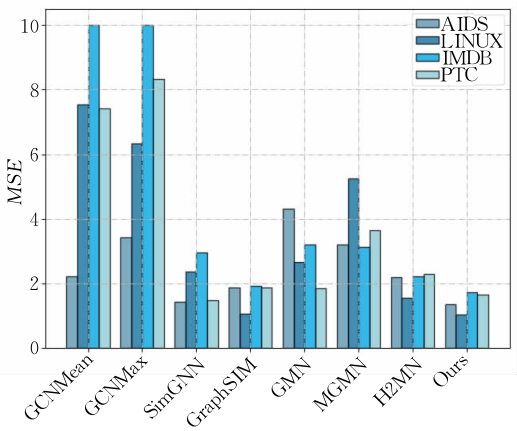


图 10　GED 预测误差对比

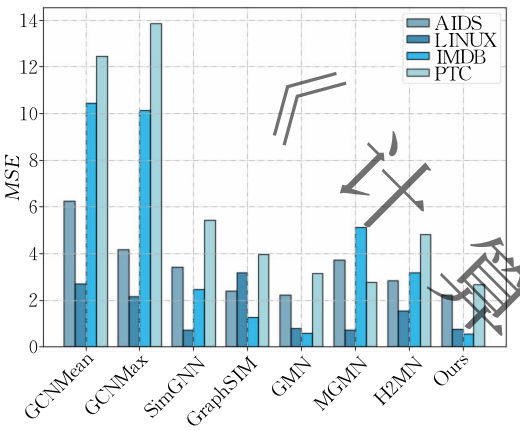


图 11　MCS 预测误差对比

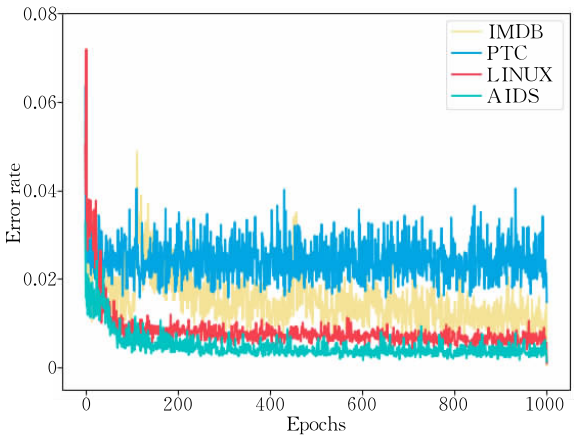


图 12　GED 预测任务损失函数变化曲线

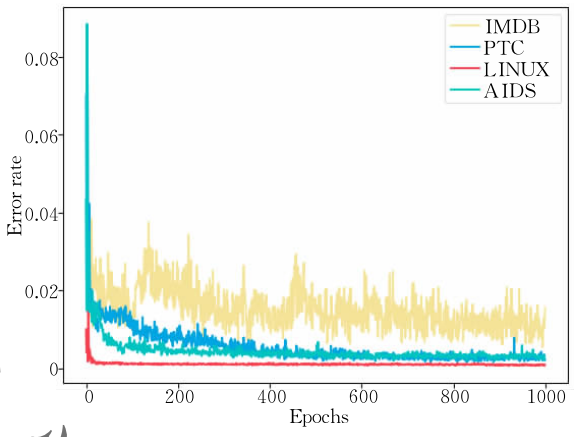


图 13　MCS 预测任务损失函数变化曲线

任务在 IMDB 上提升最多，为 7.45%. 由此推断，因大图中蕴含更为丰富的关键结构，通过子结构匹配对齐进行子图比较更能代表图的整体差异，相比之下，小图的子结构信息较匮乏并不足以反映图对之间的差异. 因此，本文方法更适用于子结构丰富的大图；③ 在 GED 预测任务中，本文模型在部分数据集的 ρ 或 τ 指标上仅取得次优效果，可以得出在该种情况下，模型的排名性能不及 GraphSIM 或 MGMN 模型，但是能够得到更高的准确性，这可能是因为结构匹配对齐时引入的随机性；在 MCS 预测任务上，值得注意的是，本文模型在 PTC 数据集的 ρ 和 τ 指标上均没有实现最佳性能，通过对 PTC 数据集的分析发现，训练图对的 GED 值分布呈现一个正偏态分布，而大多数图对 GED 值集中在 51 左右，导致训练阶段相似图对的频率高于不相似图对的频率，进而造成了性能的降低.

此外，图 12、图 13 分别可视化了本文方法在 4 个数据集上 GED 预测、MCS 预测的 *MSE* 损失函数变化曲线. 从图中可以得知，本文模型在两个任务上对训练测试数据集均收敛，由此，本文所提方法对于图之间相似度计算的优化具有有效性.

(2) 效率评估

为了进一步验证本文方法的性能，分别统计了各个方法 GED 预测的平均运行时间并可视化，如图 14 所示. 结果表明：① 在 AIDS 数据集上，本文方法效率优于 GraphSIM、GMN、MGMN；在 LINUX 上优于 GraphSIM、GMN；在 PTC 上优于 MGMN，而在 IMDB 上取得最差效率. 综上，本文方法较多数方法而言需要更多时间，这主要是因为比较耗时的结构匹配对齐步骤以及节点排序步骤；② 本文方法在时间效率方面仅优于部分方法，这点在图规模较大的 IMDB、PTC 数据集上表现更为明显，这是由于大图所蕴含的子结构更丰富，因而子结构匹配与对齐的步骤复杂性更高，但在这两个数据集上，本文方法准确性提升最大，所以本文方法对于子结构特征丰富的大图仍有效；③ 本文方法并非明显慢于其他方法，例如在 LINUX 数据集上，分别慢于 MGMN 4.2%、H^2MN 15.6%；④ 综合来看，在所有方法中 GCNMean、GCNMax 时间效率最佳，因为其直接对图嵌入进行平均池化与最大池化而没有对图结构信息进行提取，但根据表 3、表 4，这两个方法

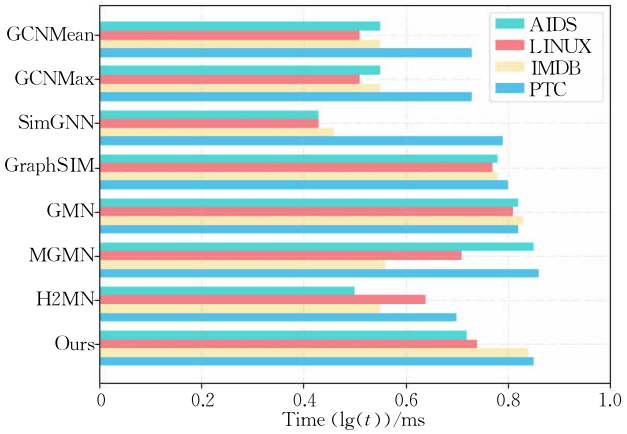


图 14 平均运行时间对比

准确性较差；⑤ 尽管本文方法时间效率不是最佳，但通过复杂性的小幅增加，换取了更好的准确性，因而可以认为其在准确性与效率之间实现了合理的权衡. 由于 MCS 预测任务上的时间对比也有类似表现，此处不再展开讨论.

除了时间效率之外，本文方法使用图对数低于其他所有基线方法. 如 4.3 节所述，初始时基于结构分布向量采用聚类策略将原始图数据划分为 k 簇，目的在于更充分地采样图对以提高模型稳健性与准确率，同时试图减少训练使用图对数. 本文分别对 4 个数据集进行了子结构提取，根据结构的分布分别聚类为{2,2,3,2}簇，设置同类图采样比例 $\alpha=70\%$，异类图采样比例 $\beta=80\%$，由此得到的图对数如表 5 所示. 由表 5 可知，使用图对数平均降低 54%，因此，本文方法在保证准确率的情况下，提升了内存使用效率与模型训练效率.

表 5 聚类后采样的图对数

数据集	图数	图对数	簇数	簇内图数	同/异类图对数	新图对数
AIDS	700	392K	2	{228,472}	137K/107K	181K
LINUX	1000	800K	2	{719,281}	297K/202K	369K
IMDB-MULTI	1500	1800K	3	{1334,130,36}	899K/226K	810K
PTC	344	94.6K	2	{244,100}	34.7K/24.4K	44K

6.3 消融实验

本节将通过消融实验来检验 MB-GSC 方法各主要组成部分的有效性，构造如下 4 个变种方法：

(1) M_1. 在 MB-GSC 方法中，删除结构提取与代表性样本生成部分，直接随机生成图对数据，以测试 GSE 与 RSG 算法的有效性；

(2) M_2. 在 MB-GSC 方法中，删除子图级比较，以测试 MSA 与 GSM 算法的有效性；

(3) M_3. 在 MB-GSC 方法中，删除图级比较，以测试图级比较过程的必要性；

(4) M_4. 在 MB-GSC 方法中，删除节点级比较，以测试节点级比较过程的必要性.

本文分别在 AIDS、IMDB、LINUX、PTC 数据集上进行 GED 预测任务，与完整算法 MB-GSC 进行对比. 实验结果如图 15 所示，实验结果显示：(1) 在 4 个数据集上，本文所提算法 MB-GSC 均取得最低误差，因此各个组成模块对模型性能均有所贡献，由此可知其有效性与必要性；(2) 综合来看，子图级比较过程对实验结果影响最大，相较于不采用子图级比较，实验结果在 AIDS、IMDB、LINUX、PTC 四个数据集上分别提升 26%、31.18%、43.73%、36.24%，其次结果差异较大的是图级比较过程，而节点比较与代表性样本生成对于实验结果影响不大；(3) 在较大数据集 IMDB 与 PTC 上，实验结果差异明显大于较小数据集 AIDS 与 LINUX，由此看出本文算法对于较大数据集性能提升更为明显，这与 6.2 节的结果分析相吻合，因而 MB-GSC 更适用于规模更大的图数据；(4) 结构提取与样本生成模块对模型性能也有一定影响，在 4 个数据集上实验结果的提升分别为 1.2%、29.23%、10.40%、8.45%. 综合上述结果分析，本文算法的各个组成模块具有有效性，对模型性能均有所贡献，其中影响最大的是子图比较过程，特别是对于较大的图，性能提升效果更好.

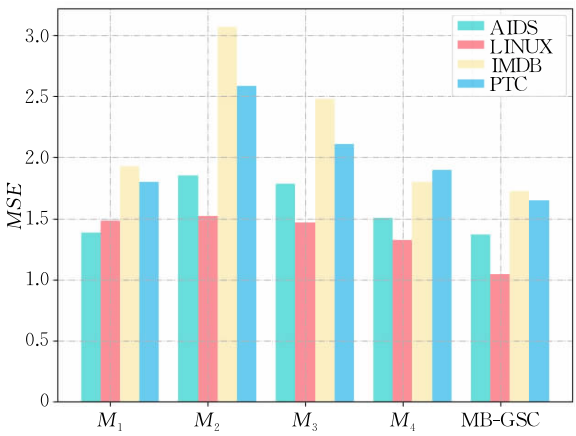


图 15 消融实验结果

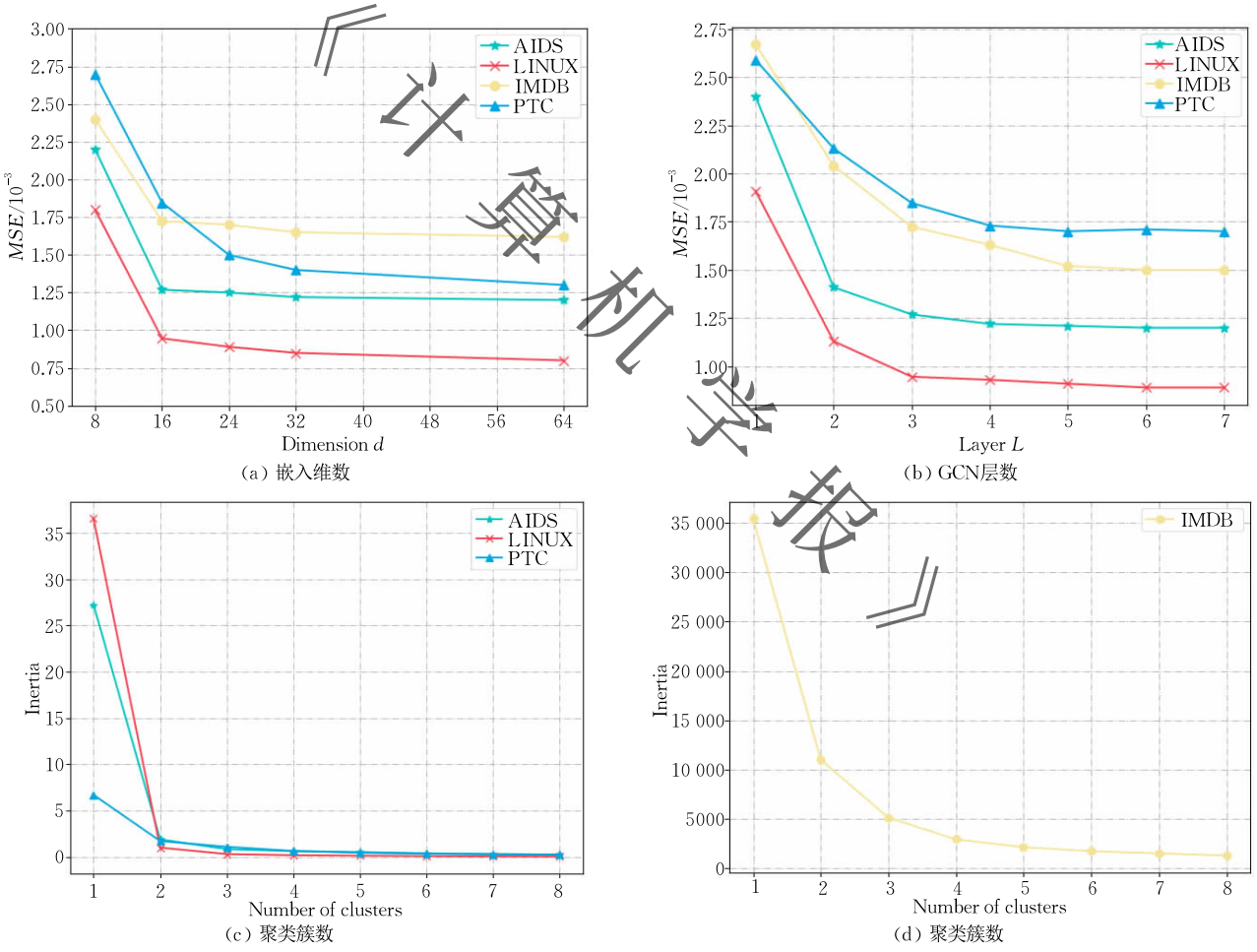
6.4 超参数实验

本节将检验本文方法中重要超参数对整体性能的影响，包括嵌入维数 d、神经网络层数 L、聚类

簇数 k，为此本文修改参数设置进行了多次实验，图 16 展示了结果，根据该图，参数设置 $d=16$、$L=3$ 时模型取得最佳性能，设置 $k=\{2,2,3,2\}$ 时聚类效果最好.

由图 16(a)可知，通过从 8 开始增加 d 的值，模型性能不断提高，并在 $d=16$ 或 24 时收敛到一个较高水平，当 d 的值继续增加时，模型性能略有下降而后逐渐稳定. 因为随着嵌入维数的增加，节点嵌入中保留了逐渐丰富的信息，当有足够数量的特征后，添加新的特征帮助并不大. 该实验结果表明，使用 $d=16$ 会获得更好的性能. 在图 16(b)中，层数 L 情况类似，为了避免过调参数，同时考虑到资源消耗，将 GCN 层数设置为 3.

另外，从图 16(c)中可以观察到对于 AIDS、LINUX、PTC 数据集，使用本文提取的元结构进行聚类时误差值较小，聚类效果较好，生成的样本更具代表性. 而图 16(d)显示，对于数据集 IMDB，使用这些结构特征进行聚类误差值非常大，并不能取得较好的聚类效果，尽管如此，本文方法仍然在此数据集上取得了较高的准确率提升，对于 GED、MCS 预测任务分别为 10.39、7.45，对此现象作以下分析：(1) IMDB 数据集中图最为稠密与复杂，本文所取 5 种元结构并不足以表示图之间的差异；(2) 聚类效果并不直接影响模型准确率，其生成的样本代表性并不好，不意味着模型准确率不高；(3) 此现象表明对于 IMDB 数据集而言，本文 MB-GSC 算法的其他模块较结构提取与样本生成模块发挥了更为重要的作用.



(a) 嵌入维数　(b) GCN层数

(c) 聚类簇数　(d) 聚类簇数

图 16　参数敏感度分析

7　总　结

为了解决现有图神经图相似度计算方法的不足，本文提出了 MB-GSC 框架. MB-GSC 通过元结构提取构造特征进行聚类并生成代表性样本，从而能够保证使用更少的数据获得平均甚至更好的效果. 其次，通过子结构匹配与对齐获取公有结构与特有结构，分别利用其形状差异与数量差异构建蕴含子结构差异信息的相似向量，并聚合了图级、节点级、子图级相似向量，使得相似向量能够表示图对之间丰富的层次比较信息而获得更高的准确率. 最后，

在多个公开数据集上进行了大量实验,实验结果表明,本文提出的 MB-GSC 框架能够有效提升图相似度计算的准确度,且在保证准确率的同时有效降低了使用样本数.

参 考 文 献

[1] Wang Zhao-Hui, Shen Hua-Wei, Cao Qi, et al. Survey on graph classification. Journal of Software, 2022, 33(1): 171-192(in Chinese)
(王兆慧, 沈华伟, 曹婧等. 图分类研究综述. 软件学报, 2022, 33(1): 171-192)

[2] Xu X, Liu C, Feng Q, et al. Neural network-based graph embedding for cross-platform binary code similarity detection//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York, USA, 2017: 363-376

[3] Wang S, Chen Z, Yu X, et al. Heterogeneous graph matching networks for unknown malware detection//Proceedings of the 28th International Joint Conference on Artificial Intelligence. Macao, China, 2019: 3762-3770

[4] Liu J, Ma G, Jiang F, et al. Community-preserving graph convolutions for structural and functional joint embedding of brain networks//Proceedings of the 2019 IEEE International Conference on Big Data. Los Angeles, USA, 2019: 1163-1168

[5] Bunke H, Allermann G. Inexact graph matching for structural pattern recognition. Pattern Recognition Letters, 1983, 1(4): 245-253

[6] Bunke H, Shearer K. A graph distance metric based on the maximal common subgraph. Pattern Recognition Letters, 1998, 19(3-4): 255-259

[7] Zeng Z, Tung A K H, Wang J, et al. Comparing stars: On approximating graph edit distance. Proceedings of the VLDB Endowment, 2009, 2(1): 25-36

[8] Xu Bing-Bing, Cen Ke-Ting, Huang Jun-Jie. A survey on graph convolutional neural network. Chinese Journal of Computers, 2020, 43(5): 755-780(in Chinese)
(徐冰冰, 岑科廷, 黄俊杰. 图卷积神经网络综述. 计算机学报, 2020, 43(5): 755-780)

[9] Bruna J, Zaremba W, Szlam A, et al. Spectral networks and locally connected networks on graphs//Proceedings of the 2nd International Conference on Learning Representations. Banff, Canada, 2014: 14-16

[10] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering//Proceedings of the 30th International Conference on Neural Information Processing Systems. Barcelona, Spain, 2016: 3844-3852

[11] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks//Proceedings of the 5th International Conference on Learning Representations. Toulon, France, 2017: 1-14

[12] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs//Proceedings of the 31st Annual Conference on Neural Information Processing Systems. Long Beach, USA, 2017: 1024-1034

[13] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks//Proceedings of the ICLR 6th International Conference on Learning Representations. Vancouver, Canada, 2018: 1-12

[14] Sanfeliu A, Fu K S. A distance measure between attributed relational graphs for pattern recognition. IEEE Transactions on Systems, 1983, SMC-13(3): 353-362

[15] Berretti S, Del B A, Vicario E. Efficient matching and indexing of graph models in content-based retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(10): 1089-1105

[16] Riesen K, Fankhauser S, Bunke H. Speeding up graph edit distance computation with a bipartite heuristic//Proceedings of the 5th International Workshop on Mining and Learning with Graphs. Firence, Italy, 2007: 21-24

[17] Fischer A, Plamondon R, Savaria Y, et al. A hausdorff heuristic for efficient computation of graph edit distance//Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition(SSPR). Berlin, Germany: Springer, 2014: 83-92

[18] Riesen K, Bunke H. Approximate graph edit distance computation by means of bipartite graph matching. Image and Vision Computing, 2009, 27(7): 950-959

[19] Neuhaus M, Riesen K, Bunke H. Fast suboptimal algorithms for the computation of graph edit distance//Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Berlin, Germany: Springer, 2006: 163-172

[20] Bai Y, Ding H, Bian S, et al. SimGNN: A neural network approach to fast graph similarity computation//Proceedings of the 12th ACM International Conference on Web Search and Data Mining. Melbourne, Australia, 2019: 384-392

[21] Bai Y, Ding H, Gu K, et al. Learning-based efficient graph similarity computation via multi-scale convolutional set matching//Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA, 2020: 3219-3226

[22] Li Y, Gu C, Dullien T, et al. Graph matching networks for learning the similarity of graph structured objects//Proceedings of the International Conference on Machine Learning. Long Beach, USA, 2019: 3835-3845

[23] Xiu H, Yan X, Wang X, et al. Hierarchical graph matching network for graph similarity computation. arXiv preprint arXiv:2006.16551, 2020

[24] Ling X, Wu L, Wang S, et al. Multilevel graph matching networks for deep graph similarity learning. IEEE Transactions on Neural Networks and Learning Systems, 2021, doi: 10.1109/TNNLS.2021.3102234

[25] Zhang Z, Bu J, Ester M, et al. H^2MN: Graph similarity learning with hierarchical hypergraph matching networks//Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. Virtual Event, Singapore, 2021: 2274-2284

[26] Bai Jiyang, Zhao Peixiang. TaGSim: Type-aware graph similarity learning and computation. Proceedings of the VLDB Endowment, 2021, 15(2): 335-347

[27] Xu H, Duan Z, Wang Y, et al. Graph partitioning and graph neural network based hierarchical graph matching for graph similarity computation. Neurocomputing, 2021, 439: 348-362

[28] Wang R, Zhang T, Yu T, et al. Combinatorial learning of graph edit distance via dynamic embedding//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, USA, 2021: 5241-5250

[29] Yang L, Zou L. Noah: Neural-optimized A* search algorithm for graph edit distance computation//Proceedings of the 2021 IEEE 37th International Conference on Data Engineering. Chania, Greece, 2021: 576-587

[30] Kashima H, Koyanagi T. Kernels for semi-structured data//Proceedings of the 19th International Conference on Machine Learning. Sydney, Australia, 2002: 291-298

[31] Borgwardt K, Kriegel H. Shortest-path kernels on graphs//Proceedings of the 5th IEEE International Conference on Data Mining. Houston, USA, 2005: 74-81

[32] Shervashidze N, Schweitzer P, van Leeuwen E J, et al. Weisfeiler-lehman graph kernels. The Journal of Machine Learning Research, 2011, 12(77): 2539-2561

[33] Shervashidze N, Vishwanathan S V N, Petri T, et al. Efficient graphlet kernels for large graph comparison//Proceedings of the 12th International Conference on Artificial Intelligence and Statistics. Clearwater Beach, USA, 2009: 488-495

[34] Coupette C, Vreeken J. Graph similarity description: How are these graphs similar?//Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. Singapore, 2021: 185-195

[35] Socher R, Chen D, Manning C D, et al. Reasoning with neural tensor networks for knowledge base completion. Advances in Neural Information Processing Systems, 2013, 1: 926-934

[36] Paszke A, Gross S, Massa F, et al. PyTorch: An imperative style, high performance deep learning library//Proceedings of the 33rd Conference on Neural Information Processing Systems. Vancouver, Canada, 2019: 8024-8035

[37] Kingma D P, Ba J. Adam: A method for stochastic optimization. arXiv preprint, arXiv:1412.6980, 2014

AN Li-Xia, M. S. candidate. Her research interests include graph neural network, graph similarity computation.

WU An-Biao, Ph. D. candidate. His research interests include graph database, graph neural network.

YUAN Ye, Ph. D., professor. His main research interests include cloud computing, bid data management (including graph data management, uncertain data management, data privacy protection) and P2P computing.

SUN Si-Qi, M. S. candidate. Her research interest is social network analysis.

WANG Guo-Ren, Ph. D., professor. His main research interests include uncertain data management, data-intensive computing, unstructured data management, distributed query processing and optimization technology.

Background

Graph similarity computation is the core step of many graph-related machine learning tasks, such as graph similarity search, graph classification. It's also a fundamental problem in computer science and its intersections such as pattern recognition, has received extensive attention in various applications. In this article, we mainly proposed a novel graph similarity computation method based on graph neural networks.

This research belongs to graph similarity computation. The current international related research is SIMGNN, which uses histogram features and neural tensor networks to model node-level and graph-level interactions, respectively.

This paper proposes MB-GSC(Meta-structure Matching and Biased Sampling based Graph Similarity Computation) for graph similarity computation. Firstly, the GSE (Graph Structure Extraction) algorithm extracts the meta-structures of the graph and construct the structure distribution vector of the graph. And then, a biased sampling strategy RSG (Representative Sample Generation) is proposed to generate representative samples based on structure distribution vector for subsequent model training. Simultaneously, the algorithm MSA (Meta Structure Alignment) is proposed to perform optimal matching and alignment of the extracted meta structures, so as to obtain the difference in shape of public structures and the number of private structures, and then construct similarity vectors of substructures containing local similarity information. Finally, the node-level pairwise comparison vector, the graph-level neural tensor network similarity vector, and the substructure similarity vector are integrated in the model to calculate the similarity of graph pairs.

This paper is supported by Foundation item: the National Natural Science Foundation of China (61932004, 62225203, U21A20516).