软件学报 ISSN 1000-9825, CODEN RUXUEW
2025,36(11):5031−5044 [doi: 10.13328/j.cnki.jos.007374] [CSTR: 32375.14.jos.007374]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于结构熵的属性图异常检测*

吴江豪[1,2], 段 亮[1,2], 岳 昆[1,2], 李昂生[3], 杨培忠[1,2]

[1](云南省智能系统与计算重点实验室 (云南大学), 云南 昆明 650500)

[2](云南大学 信息学院, 云南 昆明 650500)

[3](北京航空航天大学 计算机学院, 北京 100191)

通信作者: 岳昆, E-mail: kyue@ynu.edu.cn

摘 要: 属性图越来越多地用于描述带有关联关系的数据, 其异常检测日益受到关注. 由于属性图具有属性信息丰富、结构信息复杂等特点, 存在全局、结构和社区等多种类型的异常, 且异常特性往往隐藏于图的深度结构信息中, 现有方法仍存在结构信息丢失、异常节点检测困难等问题. 结构信息论使用编码树表示数据中的层次关系、通过最小化结构熵生成不同层次之间的关联, 可有效度量图中所蕴含的实质结构, 研究基于结构熵的属性图异常检测方法. 首先, 综合考虑属性图的结构和属性信息, 通过最小化图的结构熵, 构造属性图的 K 维编码树, 以描述其中的层次社区结构. 然后, 充分利用编码树中的节点属性和层次社区信息, 基于节点间的欧氏距离和连接程度, 设计结构异常和属性异常的评分机制, 从而确定属性图中的异常节点、检测多种类型的异常. 在多个属性图数据集上对所提方法进行对比测试, 实验结果表明, 所提方法能有效检测属性图的各类异常且显著优于现有方法.

关键词: 属性图; 异常检测; 结构信息论; 编码树; 结构熵; 层次社区结构

中图法分类号: TP311

中文引用格式: 吴江豪, 段亮, 岳昆, 李昂生, 杨培忠. 基于结构熵的属性图异常检测. 软件学报, 2025, 36(11): 5031–5044. http://www.jos.org.cn/1000-9825/7374.htm

英文引用格式: Wu JH, Duan L, Yue K, Li AS, Yang PZ. Structural-entropy-based Anomaly Detection in Attributed Graph. Ruan Jian Xue Bao/Journal of Software, 2025, 36(11): 5031–5044 (in Chinese). http://www.jos.org.cn/1000-9825/7374.htm

Structural-entropy-based Anomaly Detection in Attributed Graph

WU Jiang-Hao[1,2], DUAN Liang[1,2], YUE Kun[1,2], LI Ang-Sheng[3], YANG Pei-Zhong[1,2]

[1](Yunnan Key Laboratory of Intelligent Systems and Computing (Yunnan University), Kunming 650500, China)

[2](School of Information Science and Engineering, Yunnan University, Kunming 650500, China)

[3](School of Computer Science and Engineering, Beihang University, Beijing 100191, China)

Abstract: Attributed graphs are increasingly used to represent data with relational structures, and detecting anomalies with them is gaining attention. Due to their characteristics, such as rich attribute information and complex structural relationships, various types of anomalies may exist, including global, structural, and community anomalies, which often remain hidden within the graph's deep structure. Existing methods face challenges such as loss of structural information and difficulty identifying abnormal nodes. Structural information theory leverages encoding trees to represent hierarchical relationships within data and establishes correlations across different levels by minimizing structural entropy, effectively capturing the graph's essential structure. This study proposes an anomaly detection method for attributed graphs based on structural entropy. First, by integrating the structural and attribute information of attributed graphs, a K-dimensional encoding tree to represent the hierarchical community structure through structural entropy minimization is constructed. Next, using the

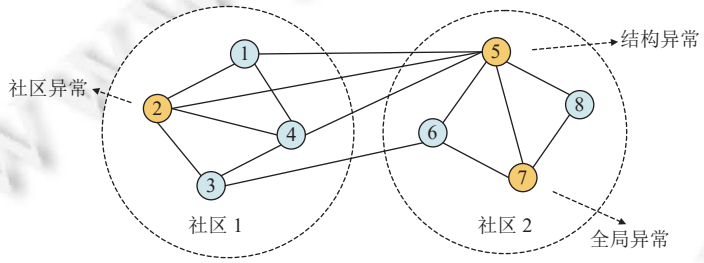
* 基金项目: 国家自然科学基金区域创新发展联合基金重点项目 (U23A20298); 云南省智能系统与计算重点实验室项目 (202405AV340009)
收稿时间: 2024-08-29; 修改时间: 2024-10-15; 采用时间: 2024-12-01; jos 在线出版时间: 2025-04-23
CNKI 网络首发时间: 2025-04-24

node attributes and hierarchical community information within the encoding tree, scoring mechanisms for detecting structural and attribute anomalies based on Euclidean distance and connection strength between nodes are designed. This approach identifies abnormal nodes and detects various types of anomalies. The proposed method is evaluated through comparative tests on several attributed graph datasets. Experimental results demonstrate that the proposed method effectively detects different types of anomalies and significantly outperforms existing state-of-the-art methods.

Key words: attributed graph; anomaly detection; structural information theory; encoding tree; structural entropy; hierarchical community structure

作为检测不符合一般规律数据的经典技术手段, 异常检测广泛用于现实中网络入侵检测、社交网络垃圾用户检测及疾病诊断等领域[1]. 早期的异常检测方法大多将用户特征作为独立矢量, 从不同维度提取用户特征或基于邻近度来判断异常[2]. 近年来, 由于属性图[3]中的节点及关系都带有类型和属性信息, 属性图成为一种包含丰富语义信息的数据组织形式, 例如, 社交网络中的用户具有“兴趣”标签、“职业”和“年龄”等重要属性, 而用户之间的交互关系可用图结构来表示. 属性图在表示数据中复杂交互关系方面的优势日益凸显, 越来越多地用于描述带有关联关系的数据, 其异常检测也日益备受关注[4], 成为图异常检测研究的重要内容.

图中密集连接的子图称为社区 (community), 属性图中自然存在多个社区, 每个节点都可被划分到相应社区, 使得社区内节点之间紧密相连. 基于社区的概念, 根据属性图中的属性和结构这两类信息, 属性图异常可分为全局异常、结构异常和社区异常[3], 如图 1 所示. 其中, 全局异常仅考虑属性信息, 是属性与图中其他节点有显著差异的节点, 节点 7 第 3 个属性值 86 与其他节点该维度的属性值有显著差异, 为全局异常节点; 结构异常是连接关系与同一社区的其他节点有显著差异的节点, 如社交网络中的垃圾信息制造者会尽量连接多个社区以发布垃圾信息, 社区 2 中节点 5 与社区 1 连接紧密, 为结构异常节点; 社区异常是属性与同一社区的其他节点有显著差异的节点, 例如, 当基因网络中的一个基因发生突变后, 其相互作用的关系虽然是正常的, 但其属性信息已发生改变, 节点 2 的第 1 个属性值明显高于所在社区中其他节点该维度的属性值, 为社区异常节点.



社区	节点	节点属性			
社区 1	1	2	21	7	67
	2	13	22	46	66
	3	4	23	47	67
	4	3	22	45	65
社区 2	5	12	23	41	64
	6	11	24	43	65
	7	13	22	86	63
	8	12	21	41	62

图 1 属性图异常

近年来, 不同的属性图异常检测方法应运而生, 早期的方法仅利用结构信息来检测异常[5], 或仅关注属性异常而忽略了结构异常[6]. 属性图中多种类型异常的检测日益受到关注, 主要采用聚类、矩阵分解等浅层学习机制[7−9], 然而, 由于属性图的结构存在维度高、特性复杂等特点, 且节点的异常特性往往隐藏在深度结构信息中, 浅层学习机制难以挖掘图的深度结构信息. 目前主流方法通过深度学习方法学习节点嵌入表示来检测异常[10−22], 但节点嵌入表示学习过程中往往存在结构信息的丢失, 从而导致难以有效检测到异常节点. 因此为了有效地推断节点的异常特性, 属性图异常检测需要处理图中多种类型的异常、并解码图的深度结构信息.

结构信息论[23]可有效解码图数据中所蕴含的实质结构, 获取图中能有效描述数据对象及其之间关联关系的信息. 结构熵是香农熵的扩展, 可在分层划分策略下度量图的不确定性, 由图中节点的度计算得到; 编码树是实质结构的编码, 可视为图的层次划分, 叶节点表示数据中的最底层元素, 非叶节点表示数据中的层次关系. 编码树将数据中的层次关系抽象为树结构, 以表示数据中不同层次之间的关联, 可通过最小化结构熵来生成, 即尽可能多地消除图中所包含的不确定性. 鉴于结构信息论具有理论基础坚实、可解释性好等特点, 近年来被越来越多地用于高阶结构度量、节点分类、社区结构隐私保护等图分析任务[24−26]. 从属性图生成的编码树, 可反映图的层次社区

信息, 为节点异常特性的表征提供丰富的语义信息和模型基础. 因此, 本文引入结构信息论, 研究基于结构熵的属性图异常检测方法, 重点解决编码树的构造和基于编码树的异常检测两个方面的问题.

为了从给定的属性图构造编码树, 我们综合考虑属性图存在的结构和属性信息, 首先将属性图构造成包含叶节点和根节点的一维编码树 (属性图中的节点为编码树中的叶节点), 并定义了社区合并、属性图压缩和编码树更新这 3 个操作. 使用社区合并操作按整体结构熵减少的原则, 将一维编码树的叶节点进行合并, 并增加社区节点, 从而将属性图中的节点合并为多个社区, 找出图中所隐含的浅层社区信息. 使用属性图压缩操作, 通过保留编码树的社区节点和根节点, 将浅层社区压缩为一个节点, 得到由社区节点组成的属性图, 再针对低维编码树进行社区合并, 得到深层次的社区信息. 使用编码树更新操作将压缩的属性图还原为原始图, 得到融合了属性信息且结构熵最小的 K 维编码树. 进一步从理论上分析了所构造编码树维度的有界性和信息的完备性, 为编码树构造的高效性给出了理论依据, 也为异常检测结果的有效性提供了保证.

为了基于编码树实现属性图的异常检测, 我们充分利用编码树中的节点属性和层次社区信息, 分别设计属性异常和结构异常评分机制. 具体而言, 计算叶节点属性与各层社区节点属性之间的欧氏距离, 再计算该距离与社区节点所处编码树中高度的加权平均值, 将其作为该叶节点的属性异常得分, 得分越高表明该叶节点属性与社区中其他节点属性的差异越大, 越可能是异常属性. 计算叶节点在每层社区中与社区之外节点之间连接的紧密程度, 再计算该值与社区节点所处编码树高度的加权平均值, 将其作为该叶节点的结构异常得分, 得分越高表明该叶节点与其他社区联系越紧密, 越可能是异常结构. 基于属性异常得分和结构异常得分得到综合异常得分, 进而检测属性图中的异常节点.

在多个真实数据集上对本文提出的方法进行了实验测试, 实验结果表明, 本文方法的 AUC 值比对比方法平均提升 2.5%, 其准确性优于现有属性图异常检测方法.

本文第 1 节阐述相关工作并分析国内外研究现状. 第 2 节陈述本文拟解决的问题并给出结构信息论基础知识. 第 3 节给出属性图编码树的构造方法. 第 4 节给出基于编码树的异常检测算法. 第 5 节给出实验结果和性能分析. 第 6 节总结全文并展望未来工作.

1 相关工作

图异常检测: 经典的图异常检测方法关注图的结构信息, 例如, 基于统计性特征的方法[5]从密度、权重、秩和特征值等方面学习邻域子图特征以检测异常点, 基于邻接矩阵及其变体的谱分析的方法[27,28]根据邻接矩阵提取特征向量以检测异常点, 基于结构相似性度量的方法[29]使用顶点结构和连通性作为聚类标准以检测异常点, 基于中心性度量的方法[30,31]根据图中所有节点的中心性或节点中介中心性以检测异常点, 基于邻接矩阵因数分解的方法[32]通过非负残差矩阵分解以检测异常点, 基于节点特征分布的方法[33,34]使用 k 近邻算法检测与其他节点相隔离的节点. 然而, 这些方法仅关注图的结构信息, 忽略了图中的属性信息, 导致图异常检测结果的准确性不高.

属性图异常检测: 属性图异常检测日益受到学界和业界的关注. 仅依赖结构信息往往难以反映真实场景中的异常特性, 经典的属性图异常检测方法同时考虑图的结构信息和属性信息, 基于聚类的方法[7]通过选定的节点子集及其相关属性, 检测偏离密集连接子图的异常点; 基于概率生成模型的方法[6]考虑到异常点可能缺失网络结构视图或属性视图, 通过模拟异常对属性图进行聚类以检测异常点; 基于残差分析的方法[8]通过表征属性信息的残差及其与网络信息的一致性以检测异常点; 基于矩阵分解的方法[9]将属性选择和异常检测作为一个整体, 通过过滤噪声和不相关节点属性以检测异常点. 这些方法利用浅层学习机制检测属性图的异常点, 难以捕获图的深度结构信息并有效推断图中的异常特性.

目前主流的属性图异常检测方法大多建立在深度神经网络基础之上[10], 例如, 基于自编码器的方法[13]在生成节点嵌入时最大限度地减少异常点带来的影响; 基于生成对抗网络的方法[14]通过学习数据的正常模式以检测异常点; 基于对比学习的方法[17]构建多视图对比学习网络, 捕获节点的异常信息; 基于图卷积网络的方法[22]通过学习节点表示, 计算重构误差以实现异常检测. 这些方法利用深度神经网络学习节点的嵌入向量, 但其过程中往往存

在结构信息的丢失, 导致异常检测结果的准确性不高.

基于结构熵的图分析: 为了度量图数据的复杂性及其存在的不确定性, Li 等人[23]提出了结构熵的概念、定义了图的 K 维结构信息, 近年来被用于图分析并取得了良好的表现. 例如, Luo 等人[24]基于结构熵度量图的高阶结构, 辅助 GNN 中节点嵌入维度的选择; Liu 等人[25]基于结构熵表达社区所揭示的信息量, 并采用残差熵最小化以实现社区隐藏和社区结构的隐私保护; Duan 等人[26]证明了具有最小结构熵的编码树可包含足够的节点分类信息, 通过构造编码树来增强基本视图以生成最优结构. 这些方法为基于结构熵的图分析技术研究提供了参考, 不同的是, 本文基于结构熵来解码属性图的结构信息并描述图中节点的异常特性.

2 问题陈述

本节首先介绍属性图、编码树和结构熵的概念, 然后给出本文方法的框架图.

定义 1. 属性图. 属性图表示为 $G=(V,E,F)$, 其中, V 为包含 n 个节点的集合, E 为包含 m 条边的集合, 每个节点 $v\in V$ 关联了一组维度为 y 的属性 (特征) 向量 F.

定义 2. 编码树. 给定属性图 G, T 为 G 的编码树, 满足:

(1) T 中每个节点 α 关联了一组非空节点集 $T_\alpha\subseteq V$, 且其关联的属性为 $\frac{1}{|T_\alpha|}\sum_{u\in T_\alpha}F_u$, u 为节点 α 的孩子节点. 即对于根节点 λ, T_λ 包含图中所有的节点, 关联的属性为其所有孩子节点属性的平均值; 对于叶节点 μ, T_μ 只包含图中的一个节点 v, 其关联的属性为该节点的属性.

(2) 对每个非叶节点 $\alpha\in T$, 其第 i 个孩子节点记为 $\alpha^{<i>}$, 且所有孩子节点的子集 $T_{\alpha^{<i>}}$ 不相交, 即 $T_\alpha=\bigcup_{i=1}^{\eta}T_{\alpha^{<i>}}$, η 为 α 的孩子节点个数.

定义 3. 一维结构熵. 用 d_v 表示属性图 G 中节点 v 的度, G 的一维结构熵为:

$$H^1(G)=-\sum_{v\in V}\frac{d_v}{2m}\log_2\frac{d_v}{2m} \tag{1}$$

定义 4. K 维结构熵. 给定维度不大于 K 的编码树 T, 属性图 G 的 K 维结构熵为:

$$H^K(G)=\min_T\sum_{\alpha\in T,\alpha\neq\lambda}H^T(G;\alpha) \tag{2}$$

$$H^T(G;\alpha)=-\frac{g_\alpha}{2m}\log_2\frac{D_\alpha}{D_{\alpha^-}} \tag{3}$$

其中, g_α 为 T_α 中节点与 $V\backslash T_\alpha$ 中节点相连的边数, $V\backslash T_\alpha$ 为 V 中不属于 T_α 的节点集合, D_α 为 T_α 中所有节点度之和, α^- 为 α 的父节点.

本文的异常检测方法表示为 $f(G):G\to\{0,1\}^n$, 将每个节点 v 与一个异常标签关联起来, 0 和 1 分别表示节点 v 正常和异常, 进一步计算节点 v 的异常得分 $S(v)$, 得分越高, 该节点越可能是异常节点.

对此, 本文提出基于结构熵的属性图异常检测方法, 框架如图 2 所示. 首先基于结构熵最小化以构造 K 维编码树, 在解码图结构信息的同时融入图的属性信息, 形成属性图的层次社区结构; 然后基于编码树提供的层次社区信息设计异常评分机制, 衡量节点的异常特性.

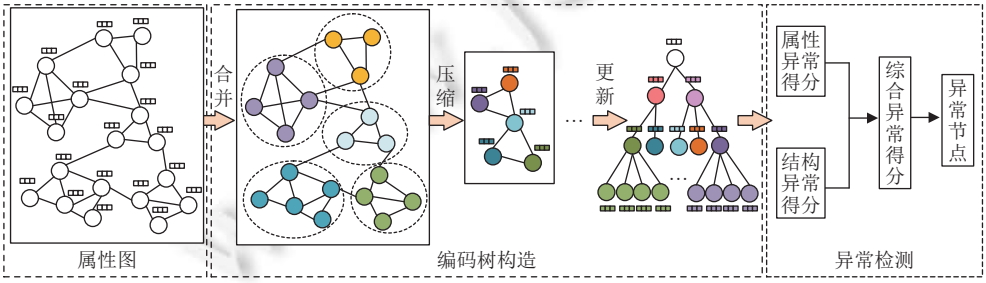


图 2 本文方法框架图

3 本文方法

3.1 编码树构造

根据前述定义, 属性图 G 的 K 维编码树可通过如下方式构造:

$$\begin{cases} T^* = \arg\min\limits_{T} H^T(G) \\ \text{s.t. } height(T) \leqslant K \end{cases} \tag{4}$$

针对属性图的特点, 本文在构造编码树的过程中融入节点的属性信息并解码属性图的层次社区信息. 对于属性图 G, 令 $P=\{P_1,P_2,\dots,P_c\}$ 为节点集合 V 的一个划分, 其中 $P_i \subseteq V$ 称为一个社区.

(1) 基本操作

为了从给定的属性图中构造编码树, 综合考虑属性图存在的结构和属性信息, 首先构造包含叶节点和根节点的一维编码树 (属性图中的节点为编码树中的叶节点), 并定义社区合并、属性图压缩和编码树更新这 3 个操作.

① 社区合并操作

使用社区合并操作按整体结构熵减少的原则, 将编码树中的节点进行合并, 并增加新的社区节点, 从而将属性图中的节点合并为多个社区, 找出图中所隐含的社区信息.

定义 5. 社区合并. 给定任意两个社区 P_i 和 P_j $(1 \leqslant i < j \leqslant c)$, 合并操作符 $op_m(P_i,P_j)$ 将 P_i 和 P_j 合并到一个新的社区 P_x $(c < x \leqslant 2c-1)$, 即 $P_x = P_i \cup P_j$, 然后从 P 中删除 P_i 和 P_j. 合并后得到新的划分, 即 $P=\{P_1,\dots,P_{i-1},P_{i+1},\dots,P_{j-1},P_{j+1},\dots,P_c,P_x\}$.

根据定义 4, K 维结构熵在社区合并前后的变化表示为:

$$\Delta SE^{P}_{(i,j)}(G) = \frac{1}{2m}((D_i-g_i)\log_2 D_i + (D_j-g_j)\log_2 D_j - (D_x-g_x)\log_2 D_x + (g_i+g_j-g_x)\log_2 2m) \tag{5}$$

比较社区合并前后的 K 维结构熵, 若合并后的 K 维结构熵减小即 $\Delta SE^{P}_{(i,j)}(G) > 0$, 则将社区进行合并. 进而, 按整体结构熵减少原则实现社区合并.

② 属性图压缩操作

使用属性图压缩操作, 通过保留编码树的社区节点和根节点, 将浅层社区压缩为一个节点、将属性图压缩为规模更小的图, 得到由社区节点组成的属性图, 再针对低维编码树进行社区合并, 得到深层次的社区信息.

定义 6. 属性图压缩. 给定属性图 G 及其节点划分 P, 压缩操作符 $op_c(P)$ 采用节点 v'_i 替代社区 P_i, 并设置 v'_i 的关联属性为 $\frac{1}{|P_i|}\sum_{u\in P_i} F_u$, 其中, u 为社区 P_i 中包含的节点, $|P_i|$ 为社区中包含的节点个数.

③ 编码树更新操作

使用编码树更新操作将压缩的属性图还原为原始图, 得到融合了属性信息且结构熵最小的 K 维编码树.

定义 7. 编码树更新. 给定编码树 T、压缩的属性图 G 及其节点划分 P, 更新操作符 $op_u(T,P)$ 通过将划分 P 中的节点作为 T 的叶节点以对其进行更新, 也就是将划分 P 插入到 T 中, 并增加 T 的高度.

(2) 构造算法

基于上述 3 个基本操作, 首先将属性图 G 中的每个节点作为一个社区, 构造包含根节点和叶节点的一维编码树 T^1. 根据公式 (2) 计算每个节点的结构熵和编码树 T^1 的结构熵. 接着, 按照定义 5 执行合并操作, 计算社区合并前后结构熵的差值, 选择差值最大的两个社区进行合并, 以迭代的方式执行社区合并操作, 直到没有使差值大于 0 的社区. 然后, 按照定义 6 执行属性图压缩操作, 将社区压缩为一个节点, 同时设置其关联属性, 将 G 压缩成规模更小的图, 新的一维编码树变为包含根节点和社区节点的二层树结构. 迭代地执行合并操作和压缩操作, 直到合并成只含一个根节点的编码树 T^0. 最后, 执行编码树更新操作, 将叶节点的孩子节点作为新的叶节点加入编码树 T^0, 构造编码树 T^1、编码树树高加 1, 反复执行构造编码树 T^2, T^3,..., 从而构造 K 维编码树 T^K.

算法 1 描述了编码树的构造过程, 得到结构熵最小的编码树也实现了属性图中层次社区结构的解码, 并融入了节点的属性信息, 为节点异常特性的度量提供了模型基础. 假设属性图 G 中包含 n 个节点、编码树 T 中的社区

节点平均包含 l 个孩子节点, 算法 1 中社区合并和属性图压缩操作的执行时间为 $O(n\log_l^2 n+q)$, 编码树更新操作的执行时间为 $O(q)$, 其中 q 为合并操作生成的社区数、最坏情况的值为 $n-1$, 因此算法 1 的时间复杂度为 $O(n\log_l^2 n)$.

算法 1. K 维编码树构造.

输入: 属性图 $G=(V, E, F)$, 编码树维度 $K>1$;
输出: G 的 K 维编码树 T.

步骤:
1. 构造高度为 1 的编码树, 记为 T^0
2. **for** h=1 to K **do**
3. 图 G_h 中所有节点单独设为一个社区, 记为 P^h
4. **while** True **do**
5. $\Delta SE^{P^h}_{(i,j)}(G_h) \leftarrow H^h_{P_i}(G_h)+H^h_{P_j}(G_h)-H^h_{P_{(i,j)}}(G_h)$ //公式 (5)
6. $P_i, P_j \leftarrow \arg\max \Delta SE^{P^h}_{(i,j)}(G_h)$
7. **if** $\Delta SE^{P^h}_{(i,j)}(G_h)>0$ **then**
8. $P^h \leftarrow op_m(P_i, P_j)$, **continue** //定义 4
9. **else**
10. $G_h \leftarrow op_c(P^h)$, **break** //定义 5
11. **end if**
12. **end while**
13. **end for**
14. **for** h=0 to $K-1$ **do**
15. $T^{(h+1)} \leftarrow op_u(T^h, P^{K-h})$ //定义 6
16. **end for**
17. **return** T

(3) 编码树性质

性质① 编码树维度的有界性

编码树的维度 K 决定了编码树构造的时间开销, 定理 1 给出 K 的上界.

定理 1. 若属性图 G 构造的编码树中, 高度大于 2 的节点平均包含 l 个孩子节点, 则编码树的维度 K 满足 $K<\log_l n$.

证明: 假设上层社区节点 (即 K 维编码树中高度大于 2 的社区节点) 平均包含 l 个孩子节点, 且 $l\geqslant 2$. 假设第 1 层社区节点 (即 K 维编码树中高度为 2 的社区节点) 平均包含 L 个孩子节点, 通常大于上层社区的平均孩子节点数 (即 $L>l$). 因此, 第 1 层社区节点平均个数为 $\frac{n}{L}$, 第 2 层社区节点平均个数为 $\frac{n}{L\times l}$, 以此类推, 第 K 层社区节点平均个数为 $\frac{n}{L\times l^{(K-1)}}$.

由于第 K 层只有一个根节点, 因此 $\frac{n}{L\times l^{(K-1)}}=1$ (即 $L\times l^{(K-1)}=n$), 则 $l^{(K-1)}=\frac{n}{L}$ $\left(\text{即}K-1=\log_l\left(\frac{n}{L}\right)\right)$, 进而可得 $K=\log_l\left(\frac{n}{L}\right)+1$, 即 $K=\log_l n-\log_l L+1$. 又由于 $L>l$, 即 $\log_l L>1$, 因此可得 $K<\log_l n$. 证毕.

根据定理 1, K 值小于由 n 个节点构成的完全 l 叉树的高度, 说明编码树的构造只需较少次数的迭代即可完成, 这一性质从理论上保证了 K 维编码树构造的效率.

性质② 编码树信息的完备性

注意到, 一维编码树包含根节点和叶节点, 而每个叶节点为一个社区, 因此, 基于一维编码树只能检测出全局异常点. 然而, 由算法 1 可知, K 维编码树包含了维度小于 K 的编码树中并不存在的社区信息, 使其能从不同层次

社区中检测出社区异常和结构异常, 定理 2 给出这一性质.

定理 2. K 维编码树包含了维度小于 K 的编码树中的信息.

证明: 属性图 G 的编码树 T 中, 社区节点的属性由其所有孩子节点属性的平均值计算所得. 一维编码树 T^1 包含根节点 λ 和叶节点 $\mu=\{\mu_1,\mu_2,\ldots,\mu_n\}$, 根节点 λ 的属性为 $F_\lambda=\frac{1}{|n|}(F_{\mu_1}+\ldots+F_{\mu_n})$.

假设 K 维编码树中每个社区节点 p 平均包含 l ($l\geqslant 2$) 个孩子节点, 第 1 层社区节点的属性为 $F^1_{P_i}=\frac{1}{|l|}(F_{\mu_i}+\ldots+F_{\mu_{i+l}})$.

从所有孩子节点属性及叶节点属性两个角度, 第 2 层社区节点 (即 K 维编码树中高度为 3 的节点) 平均包含 l 个第 1 层社区节点, 其属性可分别表示为 $F^2_{P_i}=\frac{1}{|l|}(F^1_{P_i}+\ldots+F^1_{P_{i+l}})$ 和 $F^2_{P_i}=\frac{1}{|l^2|}(F_{\mu_i}+\ldots+F_{\mu_{i+l^2}})$.

以此类推, 根节点平均包含 l 个第 $K-1$ 层社区节点, 其属性可分别表示为 $F^K_\lambda=\frac{1}{|l|}(F^{K-1}_{P_1}+\ldots+F^{K-1}_{P_l})$ 和 $F^K_\lambda=\frac{1}{|l^K|}(F_{\mu_1}+\ldots+F_{\mu_n})$.

由于每个社区节点都平均包含 l 个孩子节点, 而第 K 层的平均节点个数为 $\frac{n}{l^K}$, 根节点满足 $\frac{n}{l^K}=1$, 即 $l^K=n$, 则可得到 $F^K_\lambda=\frac{1}{|n|}(F_{\mu_1}+\ldots+F_{\mu_n})$. 因此, K 维编码树根节点的属性与一维编码树根节点的属性一致, 即高维编码树包含了低维编码树的信息. 证毕.

根据定理 2, K 维编码树包含了低维编码树的信息, 因此可检测出全局异常点. 同时, 由于 K 维编码树包含了低维编码树中并不存在的社区信息, 因此可从不同层次社区中检测出社区异常点和结构异常点. 以上性质①和性质②为异常检测结果的准确性提供了理论保证.

3.2 异常检测

从编码树中的某个叶节点 μ 出发, 自底向上沿着编码树的路径一直找到根节点 λ, 路径上的节点构成节点 μ 的祖先节点集合 anc_μ, 该集合中的节点 $\alpha\in anc_\mu$ 为叶节点 μ 在不同层次上所属的社区节点, α 中包含属于一个社区的若干叶节点. 本节基于此分别给出节点结构异常和属性异常评分策略, 进而得到每个节点的综合异常得分.

(1) 结构异常得分

对于结构异常, 若节点与所属社区之外的节点有过多的连接, 则该节点更有可能是结构异常节点. 具体而言, 对叶节点 μ 的任一祖先节点 α 所形成的社区, 节点 μ 相对于 α 的结构异常定义为 μ 在每层社区中与社区之外节点之间连接的紧密程度与社区节点所处编码树高度的加权平均值 $S_s(\mu,\alpha)$, 将其作为该叶节点的结构异常得分, 得分越高表明该叶节点与其他社区联系越紧密, 越可能是异常结构.

$$S_s(\mu,\alpha)=\left(\frac{og_\mu}{d_\mu}-\frac{g_\alpha}{D_\alpha}\right)\times \mathrm{e}^{-h(\alpha)} \tag{6}$$

其中, d_μ 为节点 μ 的度, D_α 为 T_α 中所有节点度之和, og_μ 为节点 μ 与 $V\backslash T_\alpha$ 中节点相连的边数, g_α 为 T_α 中的节点与 $V\backslash T_\alpha$ 中节点相连的边数. $h(\alpha)$ 为节点 α 在编码树中的高度, 采用 $\mathrm{e}^{-h(\alpha)}$ 进行加权, 表示层次越高的社区信息对节点 μ 异常特性的影响越小. $S_s(\mu,\alpha)$ 越大, 表明该节点相对于社区内其他节点与社区外节点连接越多, 越可能是结构异常节点.

基于叶节点 μ 相对于各层次社区的异常特性, 叶节点 μ 的结构异常得分定义为:

$$S_s(\mu)=\frac{1}{|anc_\mu|}\sum\nolimits_{\alpha\in anc_\mu}S_s(\mu,\alpha) \tag{7}$$

其中, $|anc_\mu|$ 为 μ 的祖先节点个数.

根据编码树的定义, 属性图中的节点 v 对应编码树中的叶节点 μ, 因此节点 v 的结构异常得分为 $S_s(v)=S_s(\mu)$.

(2) 属性异常得分

对于属性异常, 若节点的属性与社区内其他节点的属性存在较大差异, 则该节点为社区异常节点; 若节点属性与图中所有节点的属性都存在较大差异, 则该节点为全局异常节点. 由于编码树解码了图的层次社区结构, 且其根

节点包含图中所有节点的信息, 因此, 利用编码树的层次社区结构可同时检测社区异常和全局异常. 具体而言, 叶节点 μ 相对于任一祖先节点 α 所形成的社区, 其异常得分定义为节点 μ 与 α 中关联特征向量的加权距离 $S_a(\mu,\alpha)$, 将其作为该叶节点的属性异常得分, 得分越高表明该叶节点属性与社区中其他节点属性的差异越大, 越可能是异常属性.

$$S_a(\mu,\alpha)=dis(F_\mu,F_\alpha)\times \mathrm{e}^{-h(\alpha)} \tag{8}$$

其中, h 为编码树的高度, $\mathrm{e}^{-h(\alpha)}$ 为节点 α 在编码树中所处高度的权重系数, $dis(F_\mu,F_\alpha)$ 为叶节点 μ 与祖先节点 α 所关联的 y 维特征向量之间的欧氏距离:

$$dis(F_\mu,F_\alpha)=\sqrt{\sum_{i=1}^{y}(F_\mu^i-F_\alpha^i)^2} \tag{9}$$

当祖先节点 α 为编码树的根节点时, $S_a(\mu,\alpha)$ 表示节点 μ 在全局范围内的异常程度. 真实场景中, 一个全局异常节点往往也是社区异常节点, 因此, 本文统一检测全局异常与社区异常, 以综合衡量节点的属性异常特性.

基于叶节点 μ 相对于各层次社区的异常特性, μ 的属性异常得分定义为:

$$S_a(\mu)=\frac{1}{|anc_\mu|}\sum\nolimits_{\alpha\in anc_\mu}S_a(\mu,\alpha) \tag{10}$$

因此, 节点 v 的属性异常得分为 $S_a(v)=S_a(\mu)$.

(3) 综合异常得分

由于节点的异常特性取决于结构和属性两方面的信息, 且数据中两种异常特性分布往往不均衡, 因此, 为了得到更好的检测结果, 本文对节点 v 的结构异常和属性异常得分进行加权求和, 得到节点 v 的综合异常得分.

$$S(v)=\omega\times NoL(S_s(v))+(1-\omega)\times NoL(S_a(v)) \tag{11}$$

其中, $NoL(\cdot)$ 为归一化函数, 对得到的结构异常和属性异常评分进行归一化处理. ω $(0\leqslant\omega\leqslant 1)$ 为权重参数, 可根据实际情况改变其值以解决数据中存在异常特性不均衡的问题, 若主要为结构异常, 则可通过增加 ω 的值以提高该类异常得分的权重, 否则可通过减小 ω 的值以降低该类异常得分的权重.

(4) 检测算法

算法 2 基于构造的 K 维编码树及多层社区计算节点的异常得分, 从不同层次社区和不同异常特性中筛选出隐藏在深度结构信息中的异常节点, 可有效克服现有方法检测结果不准确的问题. 对于包含 n 个节点的属性图 G, 算法 2 中寻找祖先节点和计算异常得分步骤的执行时间分别为 $\mathrm{O}(n\times K)$ 和 $\mathrm{O}(K)$, 因此算法 2 的时间复杂度为 $\mathrm{O}(n\times K)$.

算法 2. 属性图异常检测.

输入: K 维编码树 T^K;

输出: 节点 v 的异常得分 $S(v)$.

步骤:

1. **for** T^K 的每个叶节点 μ **do**
2. $anc_\mu\leftarrow\{\}$ //叶节点 v 的祖先节点集合 anc_μ
3. $q\leftarrow\mu$
4. **while** q 不是根节点 **do**
5. $m\leftarrow q$ 的父节点
6. $anc_\mu\leftarrow anc_\mu\cup\{m\}$ //将节点 m 加入集合 anc_μ
7. $q\leftarrow m$
8. **end while**
9. **end for**
10. **for** anc_μ 中的每个节点 α **do**

11. $S_s(\mu,\alpha) \leftarrow \left(\frac{og_\mu}{d_\mu} - \frac{g_\alpha}{D_\alpha}\right) \times e^{-h(\alpha)}$ //公式 (6)

12. $dis(F_\mu, F_\alpha) \leftarrow \sqrt{\sum_{i=1}^{y} (F_\mu^i - F_\alpha^i)^2}$ //公式 (9)

13. $S_a(\mu,\alpha) \leftarrow dis(F_\mu, F_\alpha) \times e^{-h(\alpha)}$ //公式 (8)

14. **end for**

15. $S_s(\mu) \leftarrow \frac{1}{|anc_\mu|} \sum_{\alpha \in anc_\mu} S_s(\mu,\alpha)$ //公式 (7)

16. $S_s(v) \leftarrow S_s(\mu)$

17. $S_a(\mu) \leftarrow \frac{1}{|anc_\mu|} \sum_{\alpha \in anc_\mu} S_a(\mu,\alpha)$ //公式 (10)

18. $S_a(v) \leftarrow S_a(\mu)$

19. $S(v) \leftarrow \omega \times NoL(S_s(v)) + (1-\omega) \times NoL(S_a(v))$ //公式 (11)

20. **return** $S(v)$

4 实 验

为了验证本文提出的属性图异常检测方法的有效性, 我们在多个包含异常节点的真实属性图数据集上进行了实验测试, 本节给出实验结果和性能分析.

4.1 实验设置

(1) 数据集

本文选取 4 个包含异常节点的真实属性图数据集, 其统计信息如表 1 所示. 其中, Enron[35]是一个描述电子邮件交互信息的属性图数据集, 发送垃圾邮件的邮箱地址被标记为异常; Reddit[36]是社交平台 Reddit 上用户在一个子论坛中的互动数据集, 被子论坛封禁的用户被标记为异常; Weibo[7]是描述微博平台上的用户交互信息的属性图数据集, 可疑用户被标记为异常; Books[37]是亚马逊的图书购买网络, 根据原始数据中的 amazon-fail 标签获得异常信息.

表 1 数据集统计信息

数据集	节点数	边数	特征数	异常量 (占比)
Enron	13533	176987	18	5 (0.04%)
Reddit	10984	168016	64	366 (3.3%)
Weibo	8405	407963	400	868 (10.3%)
Books	1418	3695	21	28 (2.0%)

(2) 对比方法

选择如下 11 个模型作为对比方法.

● 仅检测结构异常的方法 SCAN[29]: 通过结构相似性的度量, 对顶点进行聚类以识别异常点.

● 基于张量分解的方法 ANOMALOUS[9]: 在与网络拓扑紧密关联的代表性属性空间上选择代表性实例子集, 通过残差分析度量实例的正态性以识别异常点.

● 基于自编码器的方法: AdONE[11]利用对抗性学习思想进行离群感知网络嵌入; AnomalyDAE[12]通过属性和结构双自编码器进行异常检测, 捕获网络结构和节点属性之间的相互作用, 从结构和属性两个角度度量节点的重构误差.

● 基于对抗学习的方法 GAAN[14]: 基于生成器得到伪图节点, 基于样本协方差得到矩阵虚节点和实节点的潜在空间节点表示, 鉴别器识别两个有边相连节点是否来自真图, 通过度量重构误差和识别误差以检测异常点.

● 基于对比学习的方法: SL-GAD[15]将每个目标节点的不同视图作为上下文信息, 生成属性回归以捕获属性空间中的异常点, 多视图对比学习从多个子图中挖掘结构信息并捕获结构空间中的异常点; CONAD[16]通过数据增强

策略对先验知识进行建模并集成到编码器中, 解码器重构原始网络并通过重构误差以检测异常点; GRADATE[17]构建节点-子图、节点-节点、图-子图对比学习网络, 捕获子图级和节点级的异常信息; NLGAD[18]构建子图-节点和节点-节点的对比网络, 学习准确的正态性估计以检测异常点; ARISE[19]将平均节点对相似度视为子结构内节点的拓扑异常程度, 相似度越低, 内部节点出现拓扑异常的概率越高, 从而检测异常点.

● 基于领域重建的方法 GAD-NR[22]: 根据节点表示来重构节点的邻域信息, 通过度量邻域重构误差以检测异常点.

(3) 评价指标和实验环境

为了验证本文方法的有效性, 采用 ROC 曲线下面积 AUC (area under curve) 值作为评价指标, 描述真阳性率和假阳性率之间的关系; 采用模块度 (modularity)[38]作为评价本文基于编码树的社区划分结果有效性的指标, 该值越大, 社区划分结果越好. 采用 Python 3.9、PyTorch 1.13.1+CPU 实现本文的算法, 所有实验都在配置为 10700K CPU、32 GB RAM、RTX 3070 GPU、运行 Windows 11 操作系统的机器上进行. 编码树维度、Books 数据集权重 ω、Weibo 数据集权重 ω、Reddit 数据集权重 ω 和 Enron 数据集权重 ω 分别设置为 3、0、0、0.5 和 0. 对每个测试都重复超过 5 次, 取多次测试结果的平均值作为最终结果.

4.2 实验结果

(1) 有效性测试

为了测试本文方法的有效性, 固定编码树维度为 3, 对 4 个不同数据集上异常检测结果的 AUC 值进行比较, 结果如表 2 所示. 可以看出, 本文方法在 4 个数据集上的 AUC 值都高于对比方法. 相比每个数据集上的次优方法, 本文方法在 4 个数据集上的 AUC 值分别提升了 2.5%、0.5%、3.5% 和 3.5%; 相比所有对比方法的平均效果, 本文方法在 4 个数据集上的 AUC 值分别提升了 18.4%、4.72%、16.81% 和 14.42%. 以上结果表明, 本文的方法能有效地检测属性图上的异常节点.

表 2 不同方法在各数据集上的有效性评估 (AUC (%))

方法	Enron	Reddit	Weibo	Books
SCAN	52.8	50.0	63.7	49.8
ANOMALOUS	80.8	54.9	82.3	52.8
AdONE	44.5	50.4	84.6	53.6
AnomalyDAE	54.3	55.7	91.5	62.2
GAAN	73.1	55.4	92.5	54.9
SL-GAD	63.4	55.9	52.9	51.9
CONAD	71.9	56.1	85.4	52.2
GRADATE	62.2	59.2	75.9	53.5
NLGAD	65.2	60.2	78.0	54.1
ARISE	65.6	60.0	76.6	51.9
GAD-NR	80.9	58.0	87.7	65.7
Ours	**83.4**	**60.7**	**96.0**	**69.2**

(2) 编码树的有效性

为了测试编码树维度对属性图异常检测结果有效性及社区划分结果的影响, 对编码树维度为 2–6 时异常检测结果的 AUC 和模块度进行比较, 如表 3 所示. 可以看出:

● 在 Enron 和 Reddit 数据集上、编码树维度为 2 时, 在 Weibo 数据集上、编码树维度为 4 时, 在 Books 数据集上、编码树维度为 3 时, 异常检测结果的 AUC 值最大, 本文方法的异常检测效果最佳.

● 在 Enron、Reddit 和 Books 数据集上, 编码树维度为 4 时模块度最大; 在 Weibo 数据集上、维度为 3 时, 模块度最大. 因此, 编码树的维度在 2–4 的范围内, 本文方法的社区划分结果最佳, 验证了所构建的编码树对实质结构描述的有效性.

● 编码树维度过低, 容易导致所解码出的层次结构信息不足, 影响异常检测效果; 编码树维度过高, 容易产生

冗余信息且影响异常检测的效果. 同时, 编码树构造时间也随着维度的增加而增加, 因此, 权衡方法的效率及有效性, 本文将编码树的维度设置为 3.

表 3 编码树维度的影响

编码树维度	Enron		Reddit		Weibo		Books	
	AUC (%)	模块度	AUC (%)	模块度	AUC (%)	模块度	AUC (%)	模块度
K=6	82.60	0.4303	60.63	0.3754	96.06	0.5545	68.72	0.8010
K=5	82.60	0.4303	60.64	0.3754	96.10	0.5545	68.72	0.8010
K=4	82.57	**0.4303**	60.65	**0.3754**	**96.13**	0.5545	68.71	**0.8010**
K=3	83.36	0.4299	60.73	0.3752	96.00	**0.5545**	**69.18**	0.7925
K=2	**83.76**	0.4210	**60.98**	0.3673	96.01	0.5526	67.92	0.6743

同时, 为了测试编码树对属性图异常检测结果和社区划分的有效性, 将基于结构熵解码图层次社区信息的方法分别替换为层次聚类和谱聚类方法, 对不同方法在各数据集上的 AUC 值和模块度进行比较, 结果如表 4 所示. 可以看出, 基于结构熵的方法在 4 个数据集上的表现都优于基于层次聚类和谱聚类方法. 相比谱聚类方法, 本文方法在 4 个数据集上的 AUC 值分别提升了 28.32%、5.21%、22.73% 和 0.8%, 模块度分别提升了 0.3003、0.3068、0.1730 和 0.3680; 相比层次聚类方法, 本文方法 AUC 值分别提升了 27.21%、0.86%、3.49% 和 13.08%, 模块度分别提升了 0.8032、0.5544、0.0749 和 0.4242. 上述结果表明, 基于结构熵的方法能有效地解码图的层次社区结构, 有利于属性图中节点异常特性的判断.

表 4 编码树的有效性

方法	Enron		Reddit		Weibo		Books	
	AUC (%)	模块度	AUC (%)	模块度	AUC (%)	模块度	AUC (%)	模块度
Ours	**83.36**	**0.4299**	**60.73**	**0.3752**	**96.00**	**0.5545**	**69.18**	**0.7925**
谱聚类	55.04	0.0623	55.52	0.2024	73.27	0.2477	68.38	0.5007
层次聚类	56.15	0.0061	59.87	0.3005	92.51	0.0001	56.10	−0.0022

(3) 权重参数 ω 的影响

为了测试权重参数 ω 对属性图异常检测结果有效性的影响, 对本文方法在不同权重参数 ω 时的 AUC 值进行比较, 结果如图 3 所示. 可以看出, 改变权重参数 ω, 本文方法在不同数据集上有不同的表现. 在 Enron、Weibo 和 Books 数据集上, 随着 ω 的增大, 本文方法的 AUC 值逐渐降低, 说明这 3 个数据集中的异常主要为全局异常和社区异常, 而结构异常占比较少, 因此当增大结构异常检测的权重时, 本文方法异常检测的准确率逐渐降低. 而在 Reddit 数据集上, 随着 ω 的增大, 本文方法的 AUC 值先有小幅度增加, 然后慢慢减小, 说明该数据集中的结构异常和属性异常占比相对均等, 因此当 ω 为 0.5 时, 本文方法达到最佳检测效果. 在实际应用中, 可根据具体数据集中异常的特点来调整该权重参数, 以针对相应的异常类型实现有效的异常检测.

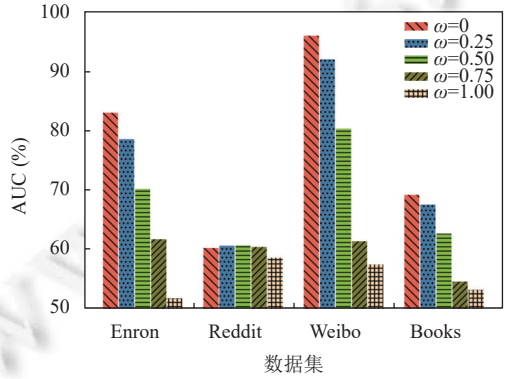


图 3 权重参数 ω 的影响

5 总结与展望

本文针对现有属性图异常检测方法存在结构信息丢失、异常检测困难等问题, 引入结构信息论以度量属性图中的结构信息, 提出了一种基于结构熵的异常检测方法, 通过建立在真实数据集上的对比测试, 验证了本文方法在属性图异常检测方面的优越性. 本文提出的属性图异常检测方法中, 编码树充分反映了属性图中的属性及结构信息, 可保证异常检测结果的准确性, 且具有良好的可解释性.

作为一种结构信息论视角下图异常检测的初步尝试, 本文方法在 Reddit 数据集上的 AUC 较对比方法提升 0.5%、在 Books 数据集上提升 3.5%, 但在不同数据集上异常检测的准确度还有提升空间, 本文方法还值得进一步优化, 并在更多的数据集上进行实验测试. 另一方面, 实际中的属性图往往表现为动态形式, 如何将本文中针对静态属性图的异常检测方法扩展到动态环境, 仍是值得深入研究的问题, 也是我们将要开展的工作.

References:

[1] Yu R, Qiu HD, Wen Z, Lin CY, Liu Y. A survey on social media anomaly detection. ACM SIGKDD Explorations Newsletter, 2016, 18(1): 1–14. [doi: 10.1145/2980765.2980767]

[2] Aggarwal CC. Outlier Analysis. New York: Springer, 2013.

[3] Ma XX, Wu J, Xue S, Yang J, Zhou C, Sheng QZ, Xiong H, Akoglu L. A comprehensive survey on graph anomaly detection with deep learning. IEEE Trans. on Knowledge and Data Engineering, 2023, 35(12): 12012–12038. [doi: 10.1109/TKDE.2021.3118815]

[4] Liu K, Dou YT, Zhao Y, Ding XY, Hu XY, Zhang RT, Ding KZ, Chen CY, Peng H, Shu K, Sun LC, Li JD, Chen GH, Jia ZH, Yu PS. BOND: Benchmarking unsupervised outlier node detection on static attributed graphs. In: Proc. of the 36th Int'l Conf. on Neural Information Processing Systems. New Orleans: ACM, 2022. 1959.

[5] Akoglu L, McGlohon M, Faloutsos C. Oddball: Spotting anomalies in weighted graphs. In: Proc. of the 14th Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining. Hyderabad: Springer, 2010. 410–421. [doi: 10.1007/978-3-642-13672-6_40]

[6] Bojchevski A, Günnemann S. Bayesian robust attributed graph clustering: Joint learning of partial anomalies and group structure. In: Proc. of the 32nd AAAI Conf. on Artificial Intelligence. New Orleans: AAAI, 2018. 2738–2745. [doi: 10.1609/aaai.v32i1.11642]

[7] Müller E, Sánchez PI, Mülle Y, Böhm K. Ranking outlier nodes in subspaces of attributed graphs. In: Proc. of the 29th Int'l Conf. on Data Engineering Workshops. Brisbane: IEEE, 2013. 216–222. [doi: 10.1109/ICDEW.2013.6547453]

[8] Li JD, Dani H, Hu X, Liu H. Radar: Residual analysis for anomaly detection in attributed networks. In: Proc. of the 26th Int'l Joint Conf. on Artificial Intelligence. Melbourne: ACM, 2017. 2152–2158.

[9] Peng Z, Luo MN, Li JD, Liu H, Zheng QH. ANOMALOUS: A joint modeling approach for anomaly detection on attributed networks. In: Proc. of the 27th Int'l Joint Conf. on Artificial Intelligence. Stockholm: ACM, 2018. 3513–3519.

[10] Li Z, Jin XL, Zhuang CZ, Sun Z. Overview on graph based anomaly detection. Ruan Jian Xue Bao/Journal of Software, 2021, 32(1): 167–193 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6100.htm [doi: 10.13328/j.cnki.jos.006100]

[11] Bandyopadhyay S, N L, Vivek SV, Murty MN. Outlier resistant unsupervised deep architectures for attributed network embedding. In: Proc. of the 13th Int'l Conf. on Web Search and Data Mining. Houston: ACM, 2020. 25–33. [doi: 10.1145/3336191.3371788]

[12] Fan HY, Zhang FB, Li ZY. AnomalyDAE: Dual autoencoder for anomaly detection on attributed networks. In: Proc. of the 2020 IEEE Int'l Conf. on Acoustics, Speech and Signal Processing. Barcelona: IEEE, 2020. 5685–5689. [doi: 10.1109/ICASSP40776.2020.9053387]

[13] Li Z, Jin XL, Wang YJ, Meng LB, Zhuang CZ, Sun Z. Anomaly node detection method based on variational graph auto-encoders in attribute networks. Pattern Recognition and Artificial Intelligence, 2022, 35(1): 17–25 (in Chinese with English abstract). [doi: 10.16451/j.cnki.issn1003-6059.202201002]

[14] Chen ZX, Liu B, Wang MQ, Dai P, Lv J, Bo LF. Generative adversarial attributed network anomaly detection. In: Proc. of the 29th ACM Int'l Conf. on Information & Knowledge Management. ACM, 2020. 1989–1992. [doi: 10.1145/3340531.3412070]

[15] Zheng Y, Jin M, Liu YX, Chi LH, Phan KT, Chen YPP. Generative and contrastive self-supervised learning for graph anomaly detection. IEEE Trans. on Knowledge and Data Engineering, 2023, 35(12): 12220–12233. [doi: 10.1109/TKDE.2021.3119326]

[16] Xu ZM, Huang X, Zhao Y, Dong YS, Li JD. Contrastive attributed network anomaly detection with data augmentation. In: Proc. of the 26th Pacific-Asia Conf. on Knowledge Discovery and Data Mining. Chengdu: Springer, 2022. 444–457. [doi: 10.1007/978-3-031-05936-0_35]

[17] Duan JC, Wang SW, Zhang P, Zhu E, Hu JT, Jin H, Liu Y, Dong ZB. Graph anomaly detection via multi-scale contrastive learning networks with augmented view. In: Proc. of the 37th AAAI Conf. on Artificial Intelligence. Washington: AAAI, 2023. 7459–7467. [doi:

10.1609/aaai.v37i6.25907]

[18] Duan JC, Zhang P, Wang SW, Hu JT, Jin H, Zhang JX, Zhou HF, Liu XW. Normality learning-based graph anomaly detection via multi-scale contrastive learning. In: Proc. of the 31st ACM Int'l Conf. on Multimedia. Ottawa: ACM, 2023. 7502–7511. [doi: 10.1145/3581783.3612064]

[19] Duan JC, Xiao B, Wang SW, Zhou HF, Liu XW. ARISE: Graph anomaly detection on attributed networks via substructure awareness. IEEE Trans. on Neural Networks and Learning Systems, 2024, 35(12): 18172–18185. [doi: 10.1109/TNNLS.2023.3312655]

[20] Zhang Z, Liu MH, Li Z, Bu JJ. Self-supervised end-to-end graph level anomaly detection. Scientia Sinica Informationis, 2023, 53(11): 2202–2213 (in Chinese with English abstract). [doi: 10.1360/SSI-2022-0179]

[21] Guo JY, Li RH, Zhang Y, Wang GR. Graph neural network based anomaly detection in dynamic networks. Ruan Jian Xue Bao/Journal of Software, 2020, 31(3): 748–762 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/5903.htm [doi: 10.13328/j.cnki.jos.005903]

[22] Roy A, Shu J, Li J, Yang C, Elshocht O, Smeets J, Li P. GAD-NR: Graph anomaly detection via neighborhood reconstruction. In: Proc. of the 17th ACM Int'l Conf. on Web Search and Data Mining. Merida: ACM, 2024. 576–585. [doi: 10.1145/3616855.3635767]

[23] Li AS, Pan YC. Structural information and dynamical complexity of networks. IEEE Trans. on Information Theory, 2016, 62(6): 3290–3339. [doi: 10.1109/TIT.2016.2555904]

[24] Luo GX, Li JX, Peng H, Yang C, Sun LC, Yu PS, He LF. Graph entropy guided node embedding dimension selection for graph neural networks. In: Proc. of the 30th Int'l Joint Conf. on Artificial Intelligence. Montreal: IJCAI, 2021. 2767–2774. [doi: 10.24963/ijcai.2021/381]

[25] Liu YW, Liu JM, Zhang ZJ, Zhu LH, LI AS. REM: From structural entropy to community structure deception. In: Proc. of the 33rd Int'l Conf. on Neural Information Processing Systems. Vancouver: NeurIPS, 2019. 1159.

[26] Duan L, Chen X, Liu WJ, Liu DL, Yue K, Li AS. Structural entropy based graph structure learning for node classification. In: Proc. of the 38th AAAI Conf. on Artificial Intelligence. Vancouver: AAAI, 2024. 8372–8379. [doi: 10.1609/aaai.v38i8.28679]

[27] Idé T, Kashima H. Eigenspace-based anomaly detection in computer systems. In: Proc. of the 10th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. Seattle: ACM, 2004. 440–449. [doi: 10.1145/1014052.1014102]

[28] Miller BA, Bliss NT, Wolfe PJ. Subgraph detection using eigenvector L1 norms. In: Proc. of the 24th Int'l Conf. on Neural Information Processing Systems. Vancouver: ACM, 2010. 1633–1641.

[29] Xu XW, Yuruk N, Feng ZD, Schweiger TAJ. SCAN: A structural clustering algorithm for networks. In: Proc. of the 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. San Jose: ACM, 2007. 824–833. [doi: 10.1145/1281192.1281280]

[30] Maulana A, Atzmueller M. Centrality-based anomaly detection on multi-layer networks using many-objective optimization. In: Proc. of the 7th Int'l Conf. on Control, Decision and Information Technologies. Prague: IEEE, 2020. 633–638. [doi: 10.1109/CoDIT49905.2020.9263819]

[31] Shashikala HBM, George R, Shujaee KA. Outlier detection in network data using the betweenness centrality. In: Proc. of the 2015 Annual IEEE Region 3 Technical, Professional, and Student Conf. Fort Lauderdale: IEEE, 2015. 1–5. [doi: 10.1109/SECON.2015.7133008]

[32] Tong HH, Lin CY. Non-negative residual matrix factorization with application to graph anomaly detection. In: Proc. of the 11th SIAM Int'l Conf. on Data Mining. Mesa: SIAM, 2011. 143–153. [doi: 10.1137/1.9781611972818.13]

[33] Bandyopadhyay S, Lokesh N, Murty MN. Outlier aware network embedding for attributed networks. In: Proc. of the 33rd AAAI Conf. on Artificial Intelligence. Honolulu: AAAI, 2019. 12–19. [doi: 10.1609/aaai.v33i01.330112]

[34] Liu FT, Ting KM, Zhou ZH. Isolation-based anomaly detection. ACM Trans. on Knowledge Discovery from Data, 2012, 6(1): 3. [doi: 10.1145/2133360.2133363]

[35] Zhao T, Deng CC, Yu KF, Jiang TW, Wang DH, Jiang M. Error-bounded graph anomaly loss for GNNs. In: Proc. of the 29th ACM Int'l Conf. on Information & Knowledge Management. ACM, 2020. 1873–1882. [doi: 10.1145/3340531.3411979]

[36] Kumar S, Zhang X, Leskovec J. Predicting dynamic embedding trajectory in temporal interaction networks. In: Proc. of the 25th ACM SIGKDD Int'l Conf. on Knowledge Discovery & Data Mining. New York: ACM. 2019. 1269–1278. [doi: 10.1145/3292500.3330895]

[37] Sánchez PI, Müller E, Laforet F, Keller F, Böhm K. Statistical selection of congruent subspaces for mining attributed graphs. In: Proc. of the 2013 IEEE Int'l Conf. on Data Mining. Dallas: IEEE, 2013: 647–656. [doi: 10.1109/ICDM.2013.88]

[38] Newman MEJ, Girvan M. Finding and evaluating community structure in networks. Physical Review E, 2004, 69(2): 026113. [doi: 10.1103/PhysRevE.69.026113]

附中文参考文献:

[10] 李忠, 靳小龙, 庄传志, 孙智. 面向图的异常检测研究综述. 软件学报, 2021, 32(1): 167–193. http://www.jos.org.cn/1000-9825/6100.htm [doi: 10.13328/j.cnki.jos.006100]

[13] 李忠, 靳小龙, 王亚杰, 孟令宾, 庄传志, 孙智. 属性网络中基于变分图自编码器的异常节点检测方法. 模式识别与人工智能, 2022, 35(1): 17–25. [doi: 10.16451/j.cnki.issn1003-6059.202201002]

[20] 张震, 刘美含, 李朝, 卜佳俊. 基于自监督的端到端图数据异常检测方法. 中国科学: 信息科学, 2023, 53(11): 2202–2213. [doi: 10.1360/SSI-2022-0179]

[21] 郭嘉琰, 李荣华, 张岩, 王国仁. 基于图神经网络的动态网络异常检测算法. 软件学报, 2020, 31(3): 748–762. http://www.jos.org.cn/1000-9825/5903.htm [doi: 10.13328/j.cnki.jos.005903]

吴江豪(1999－), 男, 硕士生, 主要研究领域为图数据分析.

李昂生(1964－), 男, 博士, 教授, 博士生导师, 主要研究领域为结构信息论.

段亮(1986－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为图数据分析.

杨培忠(1992－), 男, 博士, 副教授, CCF 专业会员, 主要研究领域为空间数据挖掘.

岳昆(1979－), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为数据与知识工程.