面向异质信息网络的双通道协同聚类算法

邱林山[1)] 房子荃[2)] 陈 璐[1)] 张天明[3)] 李天义[4)]

[1)](浙江大学计算机科学与技术学院 杭州 310027)

[2)] (浙江大学软件学院 浙江 宁波 315048)

[3)](浙江工业大学计算机科学与技术学院 杭州 310023)

[4)](奥尔堡大学计算机学院 奥尔堡 9220 丹麦)

摘 要 网络聚类广泛应用于现实世界的各个领域，受到了越来越多的关注. 由于保留了节点和链接关系的异质性，异质信息网络聚类相较于同质网络聚类具有更优的性能. 然而，现有基于图神经网络的异质信息网络聚类忽略了节点属性以及拓扑结构对聚类的权重不同的问题. 此外，这些方法仅对单一类型的目标节点聚类，而没有考虑其余类型节点的辅助作用. 为此，提出了面向异质信息网络的双通道协同聚类算法（B3C），其能够有效地融合节点属性和拓扑结构，并挖掘异质节点间的潜在相关性，从而提高聚类性能. 首先，设计了一个简单有效的双通道编码器以聚合拓扑结构及相似矩阵的邻域信息；接着，应用自训练聚类的同时学习异质信息网络表示以及优化聚类分配，并采用协同聚类机制，以对不同类型节点同时聚类；最后，利用三元中心损失（Triplet-Center Loss）学习具有区分度的节点表示，以凝聚相似节点，分离不相似节点. 在公开数据集上进行了大量实验，验证了本文提出的双通道编码器性能相较于广泛使用的图神经网络编码器有显著提升，并且 B3C 精度优于现有的基于学习的异质信息网络聚类方法.

关键词 异质信息网络；网络聚类；协同聚类；网络表示学习；图神经网络

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2023.02416

A Bi-Channel Co-Clustering Algorithm for Heterogeneous Information Networks

QIU Lin-Shan[1)] FANG Zi-Quan[2)] CHEN Lu[1)] ZHANG Tian-Ming[3)] LI Tian-Yi[4)]

[1)](*College of Computer Science and Technology, Zhejiang University, Hangzhou* 310027)

[2)](*College of Software Technology, Zhejiang University, Ningbo，Zhejiang* 315048)

[3)](*College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou* 310023)

[4)](*Department of Computer Science, Aalborg University, Aalborg* 9220 *Denmark*)

Abstract Network clustering has received increasing attention for its ubiquitous real-world applications. Heterogeneous information network (HIN) clustering improves traditional homogeneous network clustering, as HIN reserves heterogeneity of nodes and relations to enhance clustering. However, existing HIN clustering studies based on graph neural networks (GNNs) ignore different weights of node features and topology structures on clustering. Moreover, these methods only cluster target nodes of a single type, while do not consider the auxiliary of nodes of other types in HINs, which significantly degrades their performance. To this end, we propose a bi-channel co-clustering algorithm for heterogeneous information networks, abbre-

收稿日期：2022-07-28；在线发布日期：2023-03-15. 本课题得到国家自然科学基金青年项目（No. 62102351）、浙江省自然科学基金探索青年项目（No. LQ22F020018）资助. **邱林山**，博士研究生，主要研究领域为图数据处理、图机器学习. E-mail: lsqiu@zju.edu.cn. **房子荃**，博士，研究员，中国计算机学会（CCF）会员，主要研究领域为时空大数据管理与智能分析. **陈 璐**（通信作者），博士，研究员，中国计算机学会（CCF）会员，主要研究领域为数据库、大数据分析等. E-mail: luchen@zju.edu.cn. **张天明**，博士，讲师，特聘副研究员，中国计算机学会（CCF）会员，主要研究方向为图数据管理和深度学习. **李天义**，博士，助理教授，主要研究方向为时空数据管理与分析.

viated B3C, which is capable of merging node features and topology structures, as well as capturing the hidden correlations between heterogeneous nodes, in order to achieve effective HIN clustering. Specifically, we first design a simple yet effective bi-channel encoder to aggregate neighborhood information w.r.t. topology structure and a similarity matrix. Then, self-training based clustering is performed to jointly optimize the cluster assignments while learning HIN representations. Next, the co-clustering mechanism is used to cluster nodes of different types simultaneously. Finally, we adopt the triplet-center loss to obtain discriminative node embeddings, so that similar nodes are condensed and dissimilar nodes are separated. Extensive experiments on public datasets demonstrate that the designed bi-channel encoder shows significant improvements over widely used GNN encoder and B3C outperforms the state-of-the-art learning-based HIN clustering competitors.

Keywords　heterogeneous information network; network clustering; co-clustering; network representation learning; graph neural network

1　引　言

网络聚类旨在将网络中的节点划分为若干不相交簇，使得簇内节点紧密相连且属性相似. 网络聚类的应用包括但不限于社区发现、社交推荐以及异常检测，受到了学术界和工业界的广泛关注[1-5]. 现有的网络聚类研究多为同质网络设计[6-8]，即网络仅包含单一类型的节点和链接. 然而，现实世界中的系统往往由大量类型各异的对象和交互关系组成[9,10]. 这些复杂系统可以建模为异质信息网络(Heterogeneous Information Network, HIN)，以便为数据挖掘提供完整的结构信息和丰富的语义信息. 图 1(a)给出了异质信息网络实例，其包含 4 种类型节点(即作者、论文、关键字和会议)以及 3 种类型边(即论文-会议、论文-作者和论文-关键字)，以表示不同的语义关系. 从中可以得知，作者 A_1 和 A_2 是论文 P_1 的共同作者，且 P_1 发表于会议 C_1. 异质信息网络的节点通常带有属性. 例如，在图 1(a)所示的异质信息网络中，作者通常具有研究领域或所属机构等属性，论文则与关键词或摘要等属性相关联. 尽管考虑不同类型节点之间的相关性及其附加属性有利于网络挖掘任务[11,12]，很少有研究将这些观察应用于异质信息网络聚类.

为了对异质信息网络聚类，一个直接的解决方案是忽略其异质性，并应用现有的同质网络聚类算法. 该方法无法挖掘隐藏在异质信息网络中丰富的语义关系的潜在相关性. 为了充分利用异质信息网络中丰富的语义信息，同时应对其异质性带来的挑战，已存在相关工作研究了异质信息网络聚类问题[13-15]. 然而，这些方法存在以下三个缺点. 首先，它们忽

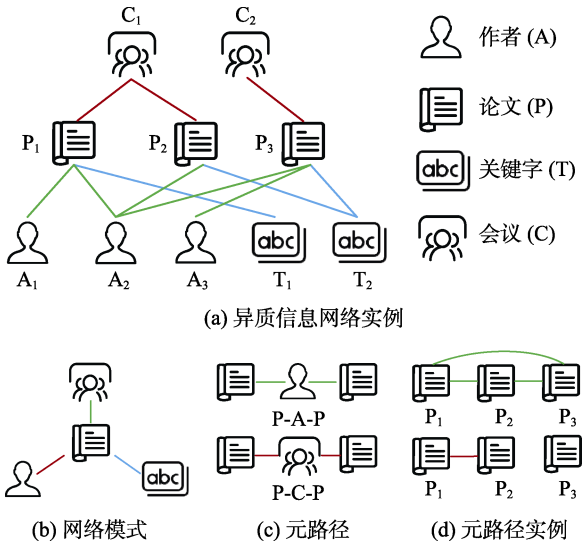


图 1　异质信息网络示例

略了节点属性[13]或简单地将节点属性和结构信息进行融合[14,15]，没有考虑异质信息网络中节点属性和拓扑结构对聚类的不同贡献度；其次，它们均对单一类型的目标节点进行聚类，忽略了异质信息网络中其它类型节点的辅助作用；最后，它们无法学习具有区分度的节点表示. 基于此，本文重新审视异质信息网络聚类，旨在充分挖掘不同类型节点之间的潜在相关性，同时对不同类型节点进行聚类以提升聚类质量. 实现上述目标主要需要解决以下三个挑战：

（1）如何有效融合异质信息网络节点属性和结构信息？现有方法[14,15]首先根据元路径构建同质网络，即元路径实例. 以图 1(a)为例，可以利用图 1(c)所示元路径 P-A-P 和 P-C-P 连接论文节点，分别表示论文间的不同语义关系. 通过该元路径，图 1(a)可以转换为如图 1(d)所示的 2 个同质网络. 接着，

利用图卷积网络(Graph Convolution Network, GCN)[16]融合节点属性和不同元路径实例以获得不同语义下的节点表示. 上述方法存在以下缺点[7,17,18]: (a)异质信息网络通常包含大量的元路径，并非所有元路径都包含丰富的语义信息，某些元路径甚至会带来噪声；(b)仅通过拓扑结构聚合节点属性，忽略属性空间的互补作用，无法有效捕获拓扑结构及节点属性的潜在相关性，造成信息的缺失. 为了解决上述问题，本文设计了一种新颖的双通道编码器，由一个属性编码器和两个拉普拉斯平滑通道组成. 具体来说，编码器首先从稀疏的原始属性向量中生成紧凑且去噪的节点表示，接着利用拉普拉斯平滑过滤器分别在结构空间(信息量最大的元路径实例)和属性空间上进行低通过滤，以分别获得节点的空间表示和属性表示. 最后，通过聚合不同空间上的节点表示以融合节点属性和结构信息. 上述方法得到的表示与任务无关，本文进一步利用自训练机制以生成聚类导向的表示.

（2）如何捕获异质信息网络中不同类型节点间的相关性？现有大多数异质信息网络聚类方法仅考虑单一类型的目标节点. 然而，本文认为考虑不同类型节点的相关性可以有效提高聚类性能. 以图1(a)为例. 一方面，论文 P_1 的共同作者(A_1 和 A_2)有很大概率属于同一个簇，因为他们研究领域相同或隶属于同一个研究机构. 另一方面，P_1、P_2 和 P_3 都由作者 A_2 撰写，它们更有可能被划分到同一个簇. 因此，对目标节点(如作者)以及辅助节点(如论文)协同聚类有利于提升聚类性能. 然而，现有方法或对单一类型节点聚类[15,19-22]，或借助矩阵分解等浅层模型以捕获异质节点间的浅层依赖[23-29]，均无法挖掘深层相关性. 基于此，本文引入协同聚类以捕获并利用其隐含的复杂相关性以指导不同类型节点的聚类过程. 具体来说，为了表示不同类型节点相关性，首先构造包含目标节点和辅助节点的二分网络，接着利用两者的节点表示以重构该二分网络.

（3）如何为异质信息网络聚类得到具有区分度的节点表示？聚类任务聚类旨在将相似的对象划分到同一个簇，而没有显式要求分离不同聚类簇，从而可能导致不同簇的重叠. 重叠部分节点难以明确其所属聚类簇. 然而，现有方法大多忽略节点表示的区分度,导致簇间难以区分[15,19-29]. 受深度度量学习的启发，本文采用了一种简单有效的损失函数，即三元中心损失[30]（Triplet-Center Loss, TCL），以使相似节点向聚类中心靠拢,并远离其它聚类中心，以有效划分不同聚类簇. 具体，本文结合自训练的聚类与三元中心损失函数，以学习任务导向且具有区分度的节点表示，从而大幅提升聚类性能.

为了解决上述挑战，本文提出了面向异质信息网络的双通道协同聚类算法 B3C. 具体而言，B3C 包含 4 个组成部分：(a)双通道编码器以捕获拓扑结构和节点属性；(b)自训练聚类模块以同时学习节点表示和优化聚类；(c)协同聚类模块以利用不同类型节点间的相关性聚类多种类型节点；(d)三元中心损失函数以凝聚相似节点并分离不相似节点.

本文工作的主要贡献可以总结为以下四点：

（1）提出了面向异质信息网络的双通道协同聚类算法 B3C，以高效地对异质信息网络进行聚类.

（2）设计了简单有效的双通道编码器以有效融合节点属性以及结构信息. 基于此，本文利用自训练机制联合优化表示学习以及聚类过程，以获得聚类导向的节点表示.

（3）捕获了不同类型节点间的相关性以提高聚类质量，并借助度量学习技术以生成针对聚类的具有区分度节点表示.

（4）在公开数据集上进行了大量的实验评估. 定量和定性实验分析验证了 B3C 相比于现有的聚类方法，具有最优的聚类性能.

接下来,本文第 2 节阐述网络聚类的相关工作；第 3 节介绍异质信息网络相关概念和形式化问题定义；第 4 节详细阐述异质信息网络聚类算法 B3C；第 5 节通过详尽的实验评估分析 B3C 的性能；第 6 节总结全文.

2 相关工作

本节简要回顾同质信息网络聚类和异质信息网络聚类的相关工作.

2.1 同质信息网络聚类

同质信息网络聚类方法主要分为三类. 第一类方法侧重于利用各种浅层范式发现稠密子图，以使簇内连边数大于簇间连边. 代表方法以基于模块度优化的聚类方法[31-33]为主. Clauset 等人[31]提出了一种贪心算法以优化模块度计算. Newman 等人[32]以矩阵形式重新定义了模块度，并在其上应用谱算法以进行聚类. Blondel 等人[33]设计了一种贪心算法，其迭代地将每个节点划分到使模块度增益最大的簇. 尽管此类方法可以发现具有高度交互关系的簇，但其计算的时空开销高昂. 此外，他们忽略了节点间属性相似性，从而限制了其有效性.

为了降低算法时空复杂度，第二类方法使用网

络嵌入技术将网络映射到低维向量空间，同时保留节点属性和网络结构信息. Deepwalk[34]将随机游走获得的节点序列视作句子，借助自然语言处理中的 skip-gram 模型[35]以学习节点表示. Node2Vec[36]通过有偏随机游走权衡广度优先搜索以及深度优先搜索的重要性，以保留网络的社区结构（同质性）和节点角色（结构等价性）. 在得到节点表示后，可以直接应用 *k*-means[37]等算法进行聚类. 然而，上述方法在学习节点表示时没有考虑社区信息. 为了解决该问题，M-NMF[38]引入模块度约束项以将微观结构和社区结构编码为统一的节点表示. 除了学习节点表示，ComE[39]将社区表示视作多元高斯分布以学习社区表示. 其借助节点表示以辅助社区发现，同时利用社区表示以学习社区感知的节点表示. 文献[40,41]在编码节点属性时亦考虑了社区信息. 在学习节点表示时考虑社区结构很大程度上提升了社区相关任务的性能，例如聚类. 本质上，这些浅层模型缺乏挖掘非线性以及复杂相关性的能力.

第三类方法利用深度学习技术为聚类任务学习节点表示，如变分自编码器[42]，去噪自编码器[43]以及对抗模型[44]等. 然而，这些模型通常是两段式的，即先学习节点表示，再应用聚类算法对其聚类. 由于表示学习与聚类过程的分离，上述方法无法取得最优的聚类结果. 受深度嵌入聚类启发[45,46]，提出了 DAEGC[47]和 SDCN[8]以学习图上面向聚类的节点表示. 其通过最小化目标分布和节点表示间的 KL 散度联合优化表示学习以及网络聚类. 相比浅层模型，深度模型能够有效发现非线性关系，因此能取得更好的聚类结果.

简而言之，上述方法聚焦于同质网络，因此难以直接有效地应用于异质信息网络，正如后续实验所验证.

2.2　异质信息网络聚类

由于异质信息网络能捕获节点以及关系间的异质性，一些工作研究了异质信息网络上的聚类问题. 本文将这些方法分为两类：非聚类导向[12-14]以及聚类导向的方法[15,19-29]. Metapath2vec[13]通过元路径随机游走采样节点上下文，并应用异质 skip-gram 模型学习节点表示. 然而，其只能编码结构信息. HetGNN[12]结合节点的多模属性(如文本和图片)学习节点表示. 其首先通过重启随机游走为每个节点采样固定数量的强关联异质邻居，并按类型聚合节点属性，最后融合不同类型表示获得统一节点表示. HetGNN 本质上学习的是多模态表示. DMGI[14]利用图互信息最大化[48]学习特定元路径的节点表示，并通过一致性正则化聚合不同元路径的表示以获得统一表示. 上述方法中表示学习与聚类分离，降低了聚类性能.

表 1　常用符号含义

符号	含义
V, E	节点集合与边集合
v_i, e_{ij}	第 i 个节点及其与节点 j 连边
$\mathcal{A}, \mathcal{R}$	节点类型和边类型集合
$\boldsymbol{A}, \widetilde{\boldsymbol{A}}, \boldsymbol{S}$	邻接矩阵、包含自环的邻接矩阵和相似矩阵
$\widetilde{\boldsymbol{D}}$	$\widetilde{\boldsymbol{A}}$ 的度矩阵
ϕ, ψ	节点和边类型映射函数
T_G	网络模式
$\boldsymbol{X}, \boldsymbol{H}, \boldsymbol{Z}$	节点属性、隐藏状态和节点表示
$\boldsymbol{x}_i, \boldsymbol{h}_i, \boldsymbol{z}_i$	节点 v_i 的原始属性、隐藏状态和节点表示
$\hat{\boldsymbol{X}}, \hat{\boldsymbol{A}}$	重构的节点属性和邻接矩阵
$\boldsymbol{W}, \hat{\boldsymbol{W}}$	可学习权重矩阵
$\boldsymbol{b}, \hat{\boldsymbol{b}}$	可学习偏置向量
$\{y_i\}$	聚类簇
$\{\boldsymbol{u}_j\}$	聚类中心
λ, γ, β	平衡系数

聚类导向方法则直接对聚类任务进行优化. Li 等人[19]提出了异质信息网络半监督聚类方法 SCHAIN. SCHAIN 考虑了节点间的属性相似性和节点间连通性的相似性，并人工选择必然存在的边以及必然不存在的边作为监督约束项进行聚类优化. 最近，Fan 等人[15]提出了端到端框架 One2Multi. One2Multi 选择信息量最大的元路径实例作为结构输入，并应用图卷积网络生成节点表示. 与 DAEGC[47]类似，One2Multi 利用自监督机制以学习任务导向的节点表示. 与之相比，本文设计了一个双通道编码器以有效学习节点表示，而非在融合属性和结构信息方面存在缺点的图卷积网络编码器，即仅在结构空间进行消息传递，忽略属性空间的互补作用，难以捕获结构和属性深层相关性. 文献[20-22]假设每个节点可由其它样本点线性组合表示，并且不同的元路径共享该线性组合系数矩阵. 因此，其利用不同的正则化技术学习此系数矩阵（如，高阶近似性[20,21]和对比正则化[22]），并基于此系数矩阵进行聚类. 上述方法同样忽略了节点属性间的相似性以及异质节点间的相关性. 此外，上述方法均无法学习具有区分度的节点表示.

考虑节点的异质性，一些工作研究了协同聚类[23-29]. Dhillon 等人[23]提出了基于二分谱图划分算法以同时对两种不同类型节点聚类. 文献[24-26]则

利用矩阵分解方法进行协同聚类. SOBG[27]通过在拉普拉斯矩阵上引入秩约束以学习具有 k 个连通分量的二分图(k 表示簇数),其中不同的连通分量对应不同的簇. 上述方法只能处理具有单一类型边的二分图,无法直接应用到具有多种语义关系的异质图. HMFClus[28]通过分解基于元路径的相似矩阵同时聚类不同类型节点,但是忽略了节点属性. SCCAIN[29]首先利用元路径和属性映射设计了一个考虑结构和属性相关性的度量矩阵,接着对该度量矩阵进行受限正交非负矩阵三元分解从而进行协同聚类. 尽管上述方法考虑了异质节点间的相关性,但其主要基于浅层模型,缺乏挖掘深层次相关性的能力. 为此,本文设计了基于深度学习的双通道编码器以提高洞悉复杂相关性的能力.

3 相关概念

本节介绍异质信息网络聚类的相关基本概念,并形式化问题定义. 表 1 总结了本文常用符号.

定义 1. 异质信息网络(Heterogeneous Information Network,HIN). 给定异质信息网络 $G=(V, E, \boldsymbol{X}, \phi, \psi)$,其中 V 和 E 分别表示节点集合与边集合;$\boldsymbol{X}$ 为节点属性矩阵,$\boldsymbol{x}_i=\mathbb{R}^d$ 为节点 v_i 属性向量,其中,d 为属性向量维度;$\phi: V\rightarrow\mathcal{A}$ ($\psi: E\rightarrow\mathcal{R}$) 是节点(边)类型映射函数,其将每个节点(每条边)映射到 $\mathcal{A}$ ($\mathcal{R}$)中特定类型,$\mathcal{A}$ 和 $\mathcal{R}$ 分别表示节点类型和边类型集合,且 $|\mathcal{A}|+|\mathcal{R}|>2$.

定义 2. 网络模式(Network Schema). 给定异质信息网络 $G=(V, E, \boldsymbol{X}, \phi, \psi)$,其网络模式记作 $T_G=(\mathcal{A},\mathcal{R})$,表示不同类型节点之间关系的元模板或模式图.

图 1(a)所示异质信息网络实例包含 4 种类型节点(即,$\mathcal{A}=\{\mathrm{P,A,C,T}\}$)以及 3 种类型边(即,$\mathcal{R}=\{\mathrm{P\text{-}A,\ P\text{-}C,\ P\text{-}T}\}$). 该异质信息网络的网络模式如图 1(b)所示,表示论文由作者撰写并发表于某一会议,且与特定关键词相关联.

定义 3. 元路径(Meta-path). 元路径是定义在网络模式 $T_G=(\mathcal{A},\mathcal{R})$ 上的路径,记作 $A_1\xrightarrow{R_1}A_2\xrightarrow{R_2}\cdots\xrightarrow{R_l}A_{l+1}$(或 $A_1-A_2-\cdots-A_{l+1}$),表示对象 A_1 和 A_{l+1} 间的复合关系 $R=R_1\circ R_2\circ\ldots\circ R_l$,其中 $\circ$ 表示关系的复合操作.

基于图 1(b)的网络模式可以定义图 1(c)所示元路径 P-A-P 和 P-C-P,用于表示不同的语义关系,即拥有共同作者或发表于同一会议的论文. 图 1(d)展示了根据不同元路径创建的元路径实例. 例如,给定元路径 P-A-P,P_2 和 P_3 均为 P_1 邻居节点. 而给定元路径 P-C-P,P_1 仅有邻居节点 P_2. 基于上述概念,下面形式化本文所研究问题.

定义 4. 异质信息网络协同聚类(HIN Clustering). 给定异质信息网络 $G=(V, E, \boldsymbol{X}, \phi, \psi)$,目标节点类型 T 和辅助节点类型 A(对应的节点集合分别记作 V_T 和 V_A). 协同聚类旨在利用不同类型节点间的相关性对节点集合 V_T 和 V_A 同时聚类,即分别将节点集合 V_T 和 V_A 划分成若干不相交的簇,使得在相同类型下,簇内节点在结构以及属性上相似.

4 双通道协同聚类算法(B3C)

本节首先概述本文提出的双通道协同聚类算法(B3C),随后详细阐述 B3C 的不同模块.

4.1 算法概述

图 2 展示了 B3C 算法,包含 2 个数据流(针对不同类型节点),两者通过协同聚类模块相关联. 每个数据流由自编码器和聚类优化模块组成. 自编码器用于融合节点属性与网络结构信息,并通过最小化重构损失以学习任务无关的节点表示. 聚类优化模块借助自训练聚类和三元中心损失以训练聚类导向和易于区分的节点表示. 最后,B3C 利用协同聚类挖掘不同类型节点间的深度相关性.

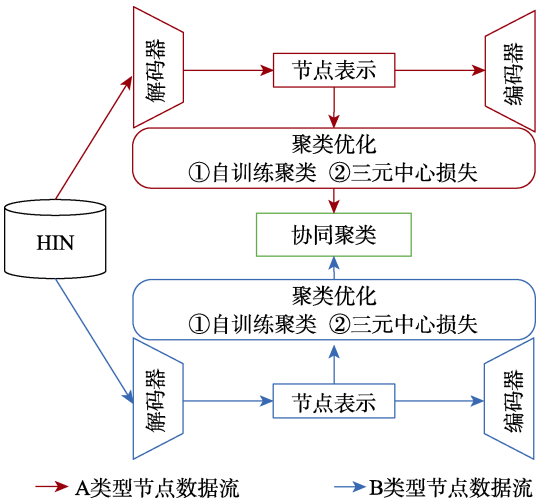


图 2 B3C 算法流程示意图

下面首先详细阐述自编码器模块,接着说明如何通过自训练和三元中心损失以学习聚类友好的节点表示. 最后,详细描述如何利用不同类型节点间的相关性以提升聚类质量.

4.2 基于双通道编码器的表示

4.2.1 编码器输入结构

在深入探讨如何将属性和结构信息编码为统一的节点表示前,首先需要明确选择哪些元路径作为

结构输入，以及如何选择这些元路径. 为了分析异质信息网络，现有研究通常根据预设元路径构建多个同质网络（元路径实例）. 通常情况下，异质信息网络包含多个不同语义的元路径. 为得到节点表示，可以设计一个多对多网络结构，为不同元路径实例各自分配一套参数，将所有选定的元路径实例作为编码器输入，并使用解码器重构所有输入. 该方案存在以下两个问题：(a)参数冗余：不同的元路径实例拥有不同的模型参数，导致训练不稳定，优化难度增加，训练时间过长；(b)表示独立：节点表示通过不同拓扑结构学习得到，它们之间几乎没有共享属性.

为了解决上述问题，本文选择信息量最大的元路径实例作为编码器输入[15]，该元路径实例决定了聚类质量. 鉴于模块度提供了衡量簇内对象紧密程度的度量指标，若模块度越高，则对象之间交互越频繁[32,33]. B3C 选择模块度最大的元路径实例作为结构输入. 为了计算元路径实例模块度，首先利用图卷积网络学习不同元路径实例的节点表示，接着使用 *k*-means 聚类，从而计算特定元路径实例的模块度.

4.2.2　双通道编码器

图卷积网络迭代地为每个节点执行消息传递以及消息聚合以获得自身表示，其定义如下式所示：

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

其中，$\boldsymbol{H}^{(l+1)}$ 为 $(l+1)$ 层输出；σ 为激活函数 $ReLU(\cdot)$；$\widetilde{\boldsymbol{A}}=\boldsymbol{A}+\boldsymbol{I}$ 为含有自环的邻接矩阵，$\boldsymbol{A}$、$\boldsymbol{I}$ 分别表示邻接矩阵和单位矩阵；$\widetilde{\boldsymbol{D}}$ 为度矩阵，其中 $\widetilde{\boldsymbol{D}}_{ii}=\sum_j \tilde{\boldsymbol{A}}_{ij}$；$\boldsymbol{W}^{(l+1)}$ 为权重矩阵.

式(1)仅通过拓扑结构进行消息传递，无法发现节点属性和拓扑结构之间某些深层相关性[17]. 在拓扑空间进行消息传递使得结构上邻近的节点具有相似表示. 然而，当节点属性与下游任务相关时，其难以发现节点在属性空间上的相关性. 因此，需要灵活地融合网络结构和节点属性信息. 此外，图卷积过滤器 $\widetilde{\boldsymbol{D}}^{-1/2}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-1/2}\boldsymbol{H}^{(l)}$ 和权重矩阵 $\boldsymbol{W}^{(l+1)}$ 的耦合将影响性能和鲁棒性[7].

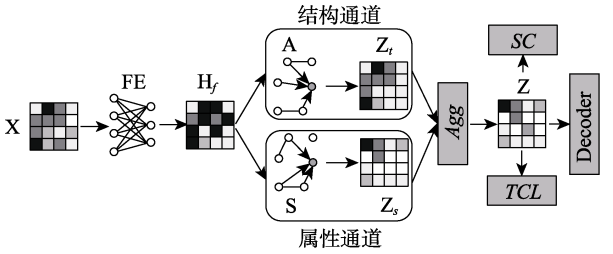
为了克服上述缺点，本文设计了一个简单有效的双通道编码器（Bi-Channel Encoder，BCE），其包含 1 个属性编码器和 2 个图卷积过滤器. 图卷积过滤器分别在结构和属性空间进行消息传递. 双通道编码器将图卷积过滤器和权重矩阵解耦. 通过不同的图卷积过滤器，双通道编码器从属性和结构的角度聚合节点表示.

双通道编码器结构如图 3 所示. 节点属性 $\boldsymbol{X}$ 经过属性编码器 FE(式(2))以无监督方式有效地提取重要属性，从而获得紧凑且去噪的潜在表示 $\boldsymbol{H}_f$. 接着，双通道编码器根据式(3)~(4)分别在结构空间(结构通道)和属性空间(属性通道)进行消息传递：

$$\boldsymbol{H}_f^{(l+1)}=\sigma(\boldsymbol{W}_f^{(l+1)}\boldsymbol{H}_f^{(l)}+\boldsymbol{b}_f^{(l+1)}) \tag{2}$$

$$\boldsymbol{Z}_t^{(l+1)}=\widetilde{\boldsymbol{A}}\boldsymbol{Z}_t^{(l)} \tag{3}$$

$$\boldsymbol{Z}_s^{(l+1)}=\boldsymbol{S}\boldsymbol{Z}_s^{(l)} \tag{4}$$



A: 邻接矩阵　　S: 相似矩阵

图 3　双通道编码器架构

其中，$\boldsymbol{W}_f^{(l+1)}$ 和 $\boldsymbol{b}_f^{(l+1)}$ 分别为编码器第 $(l+1)$ 层权重矩阵和偏置；$\boldsymbol{H}_f^{(l)}$ 为属性编码器第 l 层输出；$\boldsymbol{Z}_t^{(l+1)}$ 和 $\boldsymbol{Z}_s^{(l+1)}$ 分别为结构空间和属性空间上的节点表示，其初始化值均为 $\boldsymbol{H}_f^{(1)}$，L 为属性编码器层数；$\boldsymbol{S}$ 是由节点属性构造的相似矩阵. 构建相似矩阵最常用方法之一是连接每个节点及其 k 近邻[17]. 此方法存在以下两个缺点：(a)相似矩阵中的邻接节点可能具有完全不同的属性. 例如，当与某节点在属性上相似的节点个数少于 k 个时，仍将其与 k 个节点相连，从而带来不必要的扰动信息；(b)相似矩阵的边权相同，无法反映不同节点的重要性，造成重要信息的缺失或引入无效信息. 为了解决该问题，本文通过下式计算相似矩阵：

$$\boldsymbol{S}_{ij}=\frac{e^{(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})_{ij}}}{\sum_{k=1}^{N}e^{(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})_{ik}}} \tag{5}$$

其中，N 为节点数，$\boldsymbol{X}$ 表示原始节点属性.

由式(5)可知，属性相似性较高的节点间具有更大边权，而相似性较低的节点间边权较小，或趋于 0，因此可以自适应地根据节点相似性构建相似矩阵，无需人工设置 k 值. 在进行消息传递时，在属性空间中相似的节点被赋予较大权重，使得相似节点在低维空间中距离相近. 通过叠加多个网络层(式(3)~(4))可以聚合多跳信息.

通过式(3)~(5)可以分别得到在结构空间和属性空间上的节点表示. 接着，需要应用聚合操作(*Agg*)

以融合两者. 常用的聚合操作采用注意力机制，因其可以自适应地为结构表示和属性表示学习不同权重. 然而，注意力机制会导致训练不稳定以及增加训练时间等问题(如初始参数设置不当)[49,50]. 本文经验地使用均值聚合操作[51,52]以融合属性空间和结构空间上得到的表示以获得统一表示：

$$\boldsymbol{Z}=AVG\left(\boldsymbol{Z}_t+\boldsymbol{Z}_s\right) \tag{6}$$

4.2.3 解码器

解码器通过重构不同信息以保证编码器最大程度上融合节点属性和结构信息.

属性解码器利用节点表示重构节点属性，使得重构的属性尽可能接近原始属性：

$$\hat{\boldsymbol{H}}_f^{(l+1)}=\sigma(\hat{\boldsymbol{W}}_f^{(l+1)}\hat{\boldsymbol{H}}_f^{(l)}+\hat{\boldsymbol{b}}_f^{(l+1)}) \tag{7}$$

其中，$\hat{\boldsymbol{W}}_f^{(l+1)}$ 和 $\hat{\boldsymbol{b}}_f^{(l+1)}$ 分别为解码器第 $(l+1)$ 层权重矩阵和偏置. $\hat{\boldsymbol{H}}_f^{(0)}$ 初始化为 $\boldsymbol{Z}$. 最小化以下损失函数以重构原始属性：

$$\mathcal{L}_a=\frac{1}{2N}\sum_{i=1}^{N}\left\|\hat{\boldsymbol{x}}_i-\boldsymbol{x}_i\right\|_2^2 \tag{8}$$

其中，$\hat{\boldsymbol{x}}_i\in\hat{\boldsymbol{X}}$，$\hat{\boldsymbol{X}}=\hat{\boldsymbol{H}}_f^{(L)}$ 为重构属性，N 为节点数.

此外，为了从多个元路径实例中推断出互补信息，结构解码器利用统一表示重构每个预设元路径实例：

$$\hat{\boldsymbol{A}}_p=sigmoid(\boldsymbol{Z}\boldsymbol{W}_p\boldsymbol{Z}^{\mathrm{T}}) \tag{9}$$

其中，$\boldsymbol{Z}$ 为编码器输出；$\boldsymbol{W}_p$ 和 $\hat{\boldsymbol{A}}_p$ 分别为第 p 个元路径实例的变换矩阵和重构邻接矩阵. 为重构预设元路径实例，节点表示不仅需要尽可能地包含不同元路径实例之间的共享信息，而且具有区分不同实例的能力. 结构重构损失如下式所示：

$$\mathcal{L}_{\mathrm{s}}=\sum_{i=1}^{p}\left\|\widehat{\boldsymbol{A}}_p-\boldsymbol{A}_p\right\|_F^2 \tag{10}$$

联合式(8)和(10)，得到完整重构损失函数：

$$\mathcal{L}_r=\mathcal{L}_a+\mathcal{L}_s \tag{11}$$

4.2.4 自训练聚类

基于上述步骤学习到节点表示，可直接应用现有聚类算法（如 k-means）得到聚类结果，而不是在原始属性空间和结构空间. 由于训练过程和目标任务分离，上述步骤得到的节点表示与聚类无关，降低了聚类性能. 因此，本文同时优化表示和聚类，即自训练聚类[45,46].

自训练聚类首先计算节点表示与聚类中心相似度，此步骤称作软分配 $\boldsymbol{Q}$. 本质上，$\boldsymbol{Q}$ 表示节点归属于每个簇的概率. 该模型假设节点距离聚类中心越近，越有可能是正确的且可以作为“真值”. 其通过二次幂强化高概率的软分配重要性以获得可信赖的目标分布 $\boldsymbol{P}$. 随后，自训练聚类在生成的“真值”的监督下要求当前分布 $\boldsymbol{Q}$ 与目标分布 $\boldsymbol{P}$ 匹配，从而优化节点表示和聚类划分. 更具体地说，自训练聚类主要包含两个步骤：(a)计算软分配 $\boldsymbol{Q}$ 和目标分布 $\boldsymbol{P}$；(b)通过 KL 散度对齐 $\boldsymbol{Q}$ 和 $\boldsymbol{P}$ 以优化节点表示和聚类划分. 下面将详细阐述上述步骤细节.

B3C 使用学生 t-分布[53]度量表示 $\boldsymbol{z}_i$ (节点 v_i 表示)和聚类中心 $\boldsymbol{\mu}_j$ 的相似度：

$$\boldsymbol{Q}_{ij}=\frac{(1+\|\boldsymbol{z}_i-\boldsymbol{\mu}_j\|^2/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_{j'}(1+\|\boldsymbol{z}_i-\boldsymbol{\mu}_{j'}\|^2/\alpha)^{-\frac{\alpha+1}{2}}} \tag{12}$$

其中，$\boldsymbol{Q}_{ij}$ 可以看作将节点 v_i 划分至簇 g_j 的概率，即软分配，α 为 t-分布自由度，$\boldsymbol{\mu}_j$ 通过对节点表示应用 k-means 初始化. 直观上，式(12)计算对象和聚类中心距离.

为了通过高置信度软分配优化聚类，定义目标分布：

$$\boldsymbol{P}_{ij}=\frac{\boldsymbol{Q}_{ij}^2/f_j}{\sum_{j'}\boldsymbol{Q}_{ij'}^2/f_{j'}} \tag{13}$$

其中，$f_j=\sum_i\boldsymbol{Q}_{ij}$ 表示软分配频率之和. 式(13)使用 $\boldsymbol{Q}_{ij}^2$ 而非 $\boldsymbol{Q}_{ij}$ 以强化高置信度软分配重要性.

最后，B3C 通过最小化 KL 散度对齐软分配 $\boldsymbol{Q}$ 和目标分布 $\boldsymbol{P}$，从而对聚类进行优化：

$$\mathcal{L}_{clu}=KL(\boldsymbol{P}\|\boldsymbol{Q})=\sum_i\sum_j\boldsymbol{P}_{ij}\log\frac{\boldsymbol{P}_{ij}}{\boldsymbol{Q}_{ij}} \tag{14}$$

其中，$\boldsymbol{P}_{ij}$ 是利用 $\boldsymbol{Q}_{ij}$ 学习得到的，因此式(14)可以看作自训练过程.

为得到聚类导向的节点表示，可同时训练节点表示以及优化聚类，损失函数定义如下：

$$\mathcal{L}=\mathcal{L}_r+\lambda\mathcal{L}_{clu} \tag{15}$$

其中，$\lambda\geqslant 0$ 是平衡重构损失和聚类损失的系数. 节点 v_i 所属的簇 g_j 可通过如下函数确定：

$$y_i=\arg\max_j\boldsymbol{Q}_{ij} \tag{16}$$

4.2.5 三元中心损失函数

B3C 借助度量学习中的三元中心损失[30]，以学习具有区分度的节点表示. 三元中心损失显式要求簇内节点对应的表示向其聚类中心靠拢，并远离其它簇的聚类中心. 三元中心损失定义如下：

$$\mathcal{L}_{tc}=\sum_{i=1}^{N}\max(0,m+D(\boldsymbol{z}_i,\boldsymbol{\mu}_{y^i})-\min_{j\neq y^i}D(\boldsymbol{z}_i,\boldsymbol{\mu}_j)) \tag{17}$$

其中，m 是簇内和簇间的距离间隔，$\boldsymbol{\mu}_{y^i}$ 表示节点 u_i 对应的簇的聚类中心，$D(\cdot)$ 为平方欧氏距离：

$$D(\boldsymbol{z}_i,\boldsymbol{\mu}_{y^i})=\frac{1}{2}\|\boldsymbol{z}_i-\boldsymbol{\mu}_{y^i}\|_2^2 \tag{18}$$

直观上，$\mathcal{L}_{tc}$ 使节点表示到其聚类中心的距离和到其它聚类中心的距离之差至少为 m.

4.2.6　协同聚类

上述步骤是为单一类型节点聚类设计的，缺乏捕获不同类型节点之间的深层次相关性. 因此，B3C 对不同类型节点同时聚类以提高聚类质量. B3C 首先构造一个二分网络，其节点由所选类型节点组成，边由不同类型节点间的边组成. 接着，B3C 利用不同类型节点的表示重构该二分网络：

$$\mathcal{L}_{co}=\left\|\boldsymbol{Z}_{t_1}\boldsymbol{W}_t\boldsymbol{Z}_{t_2}^{\mathrm{T}}-\boldsymbol{A}_b\right\|_F^2 \tag{19}$$

其中，$\boldsymbol{A}_b$ 为构造的二分网络，$\boldsymbol{Z}_{t_1}$ 和 $\boldsymbol{Z}_{t_2}$ 分别为类型 t_1 和 t_2 的节点表示. 由于不同类型的节点可能被映射到不同的空间，$\boldsymbol{W}_t$ 用于将不同类型节点的表示投影到相同低维空间.

最后，B3C 完整的损失函数定义如下：

$$\mathcal{L}_o=\mathcal{L}_r+\lambda\mathcal{L}_{clu}+\beta\mathcal{L}_{tc}+\gamma\mathcal{L}_{co} \tag{20}$$

其中，λ、β 和 γ 为平衡系数. $\mathcal{L}_r$ 用于捕获输入数据的潜在表示，$\mathcal{L}_{clu}$ 用于指导训练过程以学习聚类导向的节点表示，$\mathcal{L}_{tc}$ 在最小化类内距离的同时最大化类间聚类，$\mathcal{L}_{co}$ 利用了不同类型节点间的相关性以提高聚类质量. B3C 不同部分通过公式(20)进行联合训练. 在训练过程中，针对不同类型的节点，首先分别计算 $\mathcal{L}_r$、$\mathcal{L}_{clu}$ 和 $\mathcal{L}_{tc}$，三者联合用于优化单一类型节点表示和聚类. 接着，结合 $\mathcal{L}_{co}$ 以关联不同类型节点，从而进行协同聚类.

算法 1. B3C 算法.

输入：异质信息网络 G，簇数 k，目标节点类型 T，辅助节点类型 A，类型 $T(A)$元路径 $\mathcal{MP}_T$ ($\mathcal{MP}_A$)，迭代次数 *iter*，更新区间 t，平衡系数 λ，β 和 γ

输出：聚类结果

1. 从 $\mathcal{MP}_T$ ($\mathcal{MP}_A$)中选择模块度最大的元路径 mp_{mi}^T (mp_{mi}^A)
2. 计算类型 $T(A)$的相似矩阵 S_T (S_A) //式(5)
3. 基于 mp_{mi}^T (mp_{mi}^A)和 S_T (S_A)初始化类型 $T(A)$的节点表示 Z_T (Z_A) //式(11)
4. 基于 Z_T (Z_A)，通过 *k*-means 计算初始聚类中心 μ_T (μ_A)
5. FOR　$j\leftarrow 0$ to *iter*　DO
6. 　更新 $\boldsymbol{Q}_T$ ($\boldsymbol{Q}_A$) //式(12)
7. 　IF　$j\,\%\,t=0$　THEN
8. 　　更新 $\boldsymbol{P}_T$ ($\boldsymbol{P}_A$) //式(13)
9. 　计算重构损失、聚类损失、三元中心损失和协同聚类损失 //式(11)、(14)、(17)和(19)
10. 　最小化式(20)损失函数以更新算法参数
11. 通过式(16)聚类节点 $V_T(V_A)$
12. RETURN 聚类结果

B3C 算法伪代码如算法 1 所示. 给定异质信息网络 G，簇数 k，目标节点类型 T，辅助节点类型 A，类型 $T(A)$元路径 $\mathcal{MP}_T$ ($\mathcal{MP}_A$)，迭代次数 *iter*，更新区间 t，平衡系数 λ，β 和 γ，B3C 算法输出聚类结果. 根据给定元路径集合 $\mathcal{MP}_T$ ($\mathcal{MP}_A$)，B3C 首先计算不同元路径实例的模块度，并选择模块度最大的元路径 mp_{mi}^T (mp_{mi}^A)作为结构输入(行 1). 接着，利用节点属性计算相似矩阵 S_T (S_A)(行 2). 然后，根据选定的元路径和相似矩阵利用双通道编码器初始化不同类型的节点表示 Z_T (Z_A)(行 3). 基于此，计算初始聚类中心 μ_T (μ_A)(行 4). 接下来，分别计算重构损失、聚类损失、三元中心损失和协同聚类损失，并最小化总体损失函数以更新模型参数(行 5-10). 其中，行 6-8 用于计算软分配 $\boldsymbol{Q}_T$ ($\boldsymbol{Q}_A$)和目标分布 $\boldsymbol{P}_T$ ($\boldsymbol{P}_A$). 目标分布每隔 t 次迭代更新一次以保证训练的稳定性. 最后，B3C 利用 $\boldsymbol{Q}_T$ ($\boldsymbol{Q}_A$)计算并返回聚类结果.

4.2.7　复杂度分析

令 N_T 表示类型 T 节点数，L_1 和 L_2 分别为属性编码器和图卷积层数，d_0 表示初始向量维度，$d_1,d_2,\cdots,d_{L_1}$ 表示属性编码器每层维度大小. 对于第 l 层属性编码器，其权重矩阵 $\boldsymbol{W}^{(l)}\in\mathbb{R}^{d_{l-1}\times d_l}$. 则属性编码器时间复杂度为 $\mathcal{O}_T^1(Nd_0{}^2d_1^2\cdots d_{L_1-1}{}^2d_{L_1})$. 图卷积操作时间复杂度为 $\mathcal{O}_T^2(|E|L_2+|E_T^S|L_2)$，$|E_T^S|$ 表示目标节点相似矩阵中非零元素个数. 因此，$\mathcal{O}_T^1$ 和 $\mathcal{O}_T^2$ 分别与节点数和边数呈线性关系. 对于自监督聚类模块，其时间复杂度为 $\mathcal{O}_T^3(Nk+NlogN)$，$k$ 表示簇数. 以上各项复杂度相加可得到 $\mathcal{O}_T=\mathcal{O}_T^1+\mathcal{O}_T^2+\mathcal{O}_T^3$. 类似地，对辅助类型节点 A 有 $\mathcal{O}_A=\mathcal{O}_A^1+\mathcal{O}_A^2+\mathcal{O}_A^3$. 协同聚类模块时间复杂度为 $\mathcal{O}_{co}(N_TN_Ad_{L_1}^4)$. 因此，总体时间复杂度为 $\mathcal{O}=\mathcal{O}_T+\mathcal{O}_A+\mathcal{O}_{co}$.

5　实验分析

本节在真实数据集上对本文提出的B3C算法进行实验测试，并与基准方法进行对比，以验证 B3C 的性能. 首先对本文实验设置进行说明，接着给出实验结果及分析.

5.1 实验设置

本文使用 3 个广泛使用的真实数据集进行实验测试，分别是 ACM①、DBLP②和 IMDB③，并按照节点标签划分聚类簇，即相同标签节点划分到同一簇中. 表 2 总结了评估数据集的统计数据，其中，mp.T 和 mp.A 分别表示目标节点和辅助节点的元路径，T/A 表示目标节点或辅助节点类型. ACM[54]为引用网络，节点属性为关键词词袋，论文和作者标签按研究领域划分，包括数据库，无线通信和数据挖掘；DBLP[55]为计算机科学文献数据库，节点属性为关键词词袋，作者和论文标签划分为 4 个研究领域，即数据库、数据挖掘、人工智能和信息检索；IMDB[55]为电影数据库，其中电影和导演的属性为电影情节词袋，电影根据类型分类，导演根据其所指导的电影分类，包括动作，喜剧和剧情.

表 2 测试数据集统计信息

数据集	V	E	mp.T	mp.A	T/A
ACM	Paper(3025) Author(5912) Subject(57)	P-A(9936) P-S(3025)	PAP PSP	APA APAPA APSPA	P/A
DBLP	Paper(14328) Author(4057) Conference(20) Term(7725)	P-A(19645) P-C(14328) P-T(85810)	APA APCPA APTPA	PAP PCP PTP	A/P
IMDB	Movie(4278) Actor(5257) Director(2081)	M-A(12828) M-D(4278)	MAM MDM	DMAMD	M/D

本文将提出的方法与 9 个方法进行对比，包括同质网络聚类算法以及异质网络聚类算法.

（1）k-means[37]：经典的聚类方法，直接在节点属性上进行聚类.

（2）Line[6]：网络嵌入方法，该方法保留了网络结构的一阶近似性和二阶近似性.

（3）GAE[42]：基于图卷积网络的自编码器.

（4）DAEGC[47]：在 GAE 的基础上应用 KL 散度聚类损失，以自监督的方式学习聚类导向的节点表示.

（5）AGE[7]：解耦图卷积过滤器和权重矩阵，并利用自适应学习计算聚类友好的节点表示.

（6）Mp2Vec[13]：利用基于元路径的随机游走和异质 skip-gram 模型学习节点表示.

（7）DGMI[14]：该方法将 DGI 扩展到异质多层网络，并通过最大化网络局部表示和全局表示间的互信息以学习节点表示.

（8）O2MA 和 O2MAC[15]：O2MA 利用最大信息量网络和节点属性重构多个元路径实例以学习节点表示. O2MAC 是基于 O2MA 的自监督聚类方法. 本文报告两者的最优性能. 注意到，本文提出的方法与 O2MA 主要有两个不同之处：(a)不同的编码器结构；(b)协同聚类机制.

（9）MCGC[22]：MCGC 利用对比学习得到统一的图表示，并基于该图进行聚类.

实验采用 4 个广泛使用的度量指标测试模型性能，包括准确度(ACC)、F1 分数(F1)、标准化互信息(NMI)和调整兰德系数(ARI)[44]. 分数越高，聚类性能越好.

本文使用 Pytorch 实现提出的方法，模型由 Glorot[56]初始化，并使用 Adam[57]进行训练，代码公开于 Github④. 属性编码器(解码器)维度设为 32-32，图卷积过滤器层数设为 1. λ 和 β 在区间 $[1e-2, 1e+2]$ 中搜索，γ 在区间 $[5e-4, 1e-2]$ 中搜索，学习率从 $[5e-4, 1e-2]$ 中搜索得到.

对随机游走方法(Mp2Vec)，其游走长度、游走次数和上下文大小分别设置为 100、40 和 5. 对基于 GCN 的方法(GAE、DAEGC、DMGI 和 O2MA)，其网络层维度与属性编码器相同. 对于 DAEGC、O2MA 和 B3C，本文将更新间隔设置为 5. 实验服务器配置如下：Ubuntu 20.04.2LTS，因特尔酷睿 i9-10900K 处理器，128G 内存，GeForce RTX-3090 显卡. 对于非聚类导向的方法(LINE、GAE、Mp2Vec 和 DGMI)，本文应用 k-means 对节点表示进行聚类. 为避免初始化引起的性能扰动，每个方法运行 10 次并报告平均值.

5.2 实验结果与分析

5.2.1 聚类实验结果

聚类实验结果如表 3 所示，其中粗体数值表示最优结果，下划线数值表示次优结果，ACM-P(ACM-A)表示 ACM 数据集中论文(作者)的聚类结果，DBLP 与 IMDB 类似.

首先，k-means 忽略网络结构，直接利用节点属性进行聚类，未能充分挖掘网络信息. 因此，k-means 在大多数情况下性能较差. 但是，在某些数据集上(如，ACM-P)性能比网络聚类算法优越(如，Mp2Vec)，说明了节点属性作为辅助信息，对网络聚类性能有重要作用. 其次，相比于同质网络方法(Line 和 GAE)，异质网络方法(DMGI、O2MA、MCGC 和 B3C)具有更优的聚类质量，无论其是否学习聚类

① http://dl.acm.org
② https://dblp.uni-trier.de/
③ https://www.imdb.com/
④ https://github.com/ZJU-DAILY/b3c

表 3　聚类实验结果

数据集	评价指标	*k*-means	Line	GAE	DAEGC	AGE	Mp2Vec	DMGI	O2MA	MCGC	B3C
ACM-P	ACC	0.6326	0.6446	0.7579	0.9094	0.8859	0.6241	0.7639	0.9051	0.9135	**0.9269**
	F1	0.6348	0.6579	0.7600	0.9095	0.8856	0.5914	0.7495	0.9063	0.9141	**0.9273**
	NMI	0.3015	0.3948	0.4061	0.6938	0.6490	0.4019	0.5503	0.7044	0.7149	**0.7450**
	ARI	0.2754	0.3411	0.4204	0.7515	0.6965	0.3306	0.5221	0.7437	0.7630	**0.7963**
ACM-A	ACC	0.3575	0.6601	0.6469	0.7026	0.8022	0.6625	0.8222	0.9023	0.8892	**0.9191**
	F1	0.2845	0.6651	0.6596	0.6968	0.8031	0.6674	0.8066	0.9034	0.8915	**0.9182**
	NMI	0.0018	0.4063	0.4072	0.3879	0.5016	0.4140	0.5771	0.6751	0.6564	**0.7196**
	ARI	0.0017	0.3651	0.3422	0.3833	0.4861	0.3708	0.5949	0.7173	0.6784	**0.7455**
DBLP-A	ACC	0.3871	0.8960	0.9016	0.8859	0.9043	0.9065	0.8925	0.9103	0.9178	**0.9283**
	F1	0.3152	0.8880	0.8946	0.8731	0.8918	0.9019	0.8823	0.9074	0.9157	**0.9239**
	NMI	0.1129	0.7033	0.7190	0.6985	0.7398	0.7424	0.6919	0.7282	0.7333	**0.7717**
	ARI	0.0702	0.7579	0.7656	0.7354	0.7805	0.7874	0.7530	0.7680	0.7980	**0.8262**
DBLP-P	ACC	0.3664	0.4736	0.5323	0.8173	0.6508	0.5942	0.8331	0.8267	0.8301	**0.8507**
	F1	0.2920	0.4241	0.4653	0.8063	0.6659	0.4877	0.8087	0.8009	0.8133	**0.8381**
	NMI	0.0615	0.2956	0.3445	0.5659	0.3967	0.2546	0.5665	0.5519	0.5656	**0.5800**
	ARI	0.0176	0.1708	0.2225	0.6009	0.3000	0.2798	0.6377	0.6197	0.6298	**0.6617**
IMDB-M	ACC	0.3672	0.3578	0.3800	0.4441	0.3920	0.4126	0.4774	0.4255	**0.5072**	0.4935
	F1	0.2231	0.3545	0.3457	0.4363	0.2937	0.4091	0.4821	0.4145	0.4527	**0.4907**
	NMI	0.0049	0.0031	0.0108	0.0433	0.0091	0.0289	0.0836	0.0491	**0.1350**	0.0772
	ARI	–0.0028	0.0032	0.0107	0.0510	0.0020	0.0297	0.0769	0.0600	**0.1322**	0.0842
IMDB-D	ACC	0.3730	0.4321	0.4463	**0.4825**	0.3777	0.4625	0.4569	0.3473	0.4007	0.3710
	F1	0.2132	0.4272	0.4372	**0.4578**	0.3026	0.4356	0.4532	0.3549	0.2799	0.3636
	NMI	0.0064	0.0265	0.0415	**0.0836**	0.0041	0.0504	0.0456	0.0031	0.0038	0.0048
	ARI	–0.0049	0.0297	0.0379	**0.0852**	–0.0020	0.0413	0.0419	0.0001	0.0006	0.0028

导向的节点表示. 这是因为后者能聚合多个语义信息以学习表示. 再次，相比于浅层网络表示方法(Line、Mp2Vec)，基于 GCN 的方法(GAE、DAEGC、DMGI、O2MA 和 B3C)具有更优的性能. 这是因为 GCN 不仅能捕获节点之间的非线性关系，而且考虑了节点属性. 此外，Line 在 ACM-P 上性能优于 Mp2Vec，而两者在 ACM-A 和 DBLP-A 上具有相似的性能表现. 这是因为 Line 考虑了一阶和二阶相似性，而 Mp2Vec 只考虑一阶相似性，使得其性能受限. 再者，DAEGC 在 ACM-P 上比 O2MA 性能更优. 这是因为：(a) DAEGC 在高阶邻域上进行消息传递，如一阶和二阶邻居；(b) DAEGC 使用了注意力机制衡量不同邻居的重要性. 此外，DMGI 在某些情况下性能优于 O2MA 和 MCGC(如在 DBLP-P 上). 尽管不是面向聚类任务的方法，DMGI 能够对异质信息网络全局特性进行建模，这有助于学习具有区分度的表示. 本质上，MCGC 仍然是基于图结构的聚类方法. 因此，B3C 的性能在大多数数据集上仍优于 MCGC.

最后，本文提出的算法在大多数数据集上都优于对比方法. B3C 在 ACM-A 上展现了显著的性能提升，在评价指标 NMI 和 ARI 上分别有 6.59%和 3.93%的性能提升. 相比于最优基准方法，B3C 在 ACM 和 DBLP 上分别具有 3.19%和 2.60%的平均性能提升. 这是因为 B3C 权衡了属性和结构信息对表示学习的不同贡献度，而非简单地使用图卷积网络聚合两者. 得益于本文提出的编码器，B3C 能有效融合属性和结构信息. 此外，借助自训练聚类和三元中心损失，节点表示针对聚类任务进行了优化. 最后，B3C 利用了不同类型节点的深层相关性进行协同聚类. 因此，应用深度学习技术捕获异质信息网络的复杂关系很有前景. 基于此，特定的挖掘任务能从任务驱动的学习过程中受益，例如本文所使用的基于自训练的异质信息网络聚类. 注意到，B3C 在 IMDB-D 上聚类效果一般(类似地，AGE、O2MA 和 MCGC 等)，这是由于 IMDB-D 簇内结构松散(即，模块度低)，难以形成有效的社区结构，B3C 每次消息传递时，只聚合一阶邻域信息，而 DAEGC、Line 和

Mp2Vec 等方法能够关注到更远距离的结构信息. 此外，IMDB-D 只有单个可用元路径，B3C、O2MA 和 MCGC 等方法无法利用互补信息充分挖掘信息. 当数据集簇内结构较为紧密，并且存在多个互补元路径时，B3C 能更好利用结构信息和互补信息从而达到更优的聚类性能，如 ACM 和 DBLP 数据集实验结果所示. 未来工作将进一步优化 B3C 上述缺陷，以满足不同数据集需求.

5.2.2 双通道编码器性能评估

为了评估双通道编码器性能，本文将其与两个变体(BCE_adj 和 BCE_sm)以及 GCN 编码器进行对比实验. BCE_adj 和 BCE_sm 分别在结构空间和属性空间上进行消息传递，并移除了自训练聚类、协同聚类和三元中心损失模块以消除它们的影响. 为评估有效性，本文直接应用 *k*-means 对训练后节点表示进行聚类. 为了评估效率，本文比较不同方法的收敛速度. 实验结果分别如表 4 和图 4 所示.

表 4 给出了双通道编码有效性的实验结果. (a)对比 BCE_adj 和 GCN 编码器，BCE_adj 在所有测试数据集上具有更优的聚类结果，验证了解耦权重矩阵和拉普拉斯过滤器的有效性. 属性编码器可以在一定程度上消除数据噪声，并通过权重学习压缩属性，而拉普拉斯过滤器进一步消除高频噪声，从而生成具有丰富表达能力的节点表示. (b)对比 BCE_sm 和 GCN 编码器，BCE_sm 在 ACM-A 和 ACM-P 上实验结果亦优于 GCN 编码器. 该现象说明了在相似矩阵上传递信息的重要性. (c)在多数情况下，BCE_adj 优于 BCE_sm. 这是因为 BCE_adj 融合了节点属性和结构信息学习表示，BCE_sm 仅关注节点属性，表明在拓扑结构空间传递属性性能优于相似矩阵. (d)BCE 在所有测试数据集上具有最优实验结果. 表明仅在属性空间或结构空间进行消息传递会导致信息缺失，而 BCE 能更灵活地聚合属性和结构信息.

表 4 双通道编码器有效性

数据集	模型	ACC	F1	NMI	ARI
ACM-A	GCN	0.8525	0.8497	0.5832	0.6275
	BCE_adj	0.8791	0.8782	0.6122	0.6799
	BCE_sm	0.8723	0.8719	0.6147	0.6655
	BCE	**0.8858**	**0.8827**	**0.6319**	**0.6889**
ACM-P	GCN	0.8711	0.8722	0.6203	0.6670
	BCE_adj	0.8850	0.8865	0.6495	0.6936
	BCE_sm	0.8807	0.8814	0.6442	0.6877
	BCE	**0.9015**	**0.9020**	**0.6778**	**0.7295**
DBLP-P	GCN	0.8162	0.8031	0.5510	0.6012
	BCE_adj	0.8210	0.8124	0.5631	0.6037
	BCE_sm	0.4566	0.3113	0.1041	0.0727
	BCE	**0.8359**	**0.8179**	**0.5647**	**0.6385**
DBLP-A	GCN	0.9049	0.9000	0.7163	0.7696
	BCE_adj	0.9118	0.9061	0.7370	0.7898
	BCE_sm	0.5721	0.5753	0.2882	0.1722
	BCE	**0.9201**	**0.9146**	**0.7599**	**0.8114**

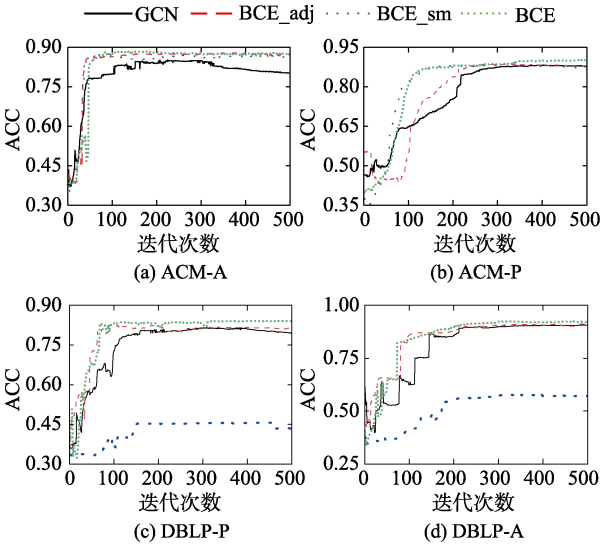


图 4 双通道编码器效率

双通道编码效率的实验结果如图 4 所示. (a)BCE_adj 和 BCE 始终比 GCN 编码器收敛快. 虽然 BCE_adj 在大多数情况下与 BCE 有相近的收敛速度，但 BCE 有更高的聚类质量. (b)由于数据复杂度的提高，相比于 ACM-A 和 ACM-P，在 DBLP-A 和 DBLP-A 上的收敛曲线波动更大，需要迭代更多轮次以达到最优结果.

综上，相比于其变体和 GCN 编码器，本文提出的编码器能更快更好地聚合属性和结构信息.

5.2.3 消融实验

本小结通过消融实验评估 B3C 算法各个模块的有效性，实验结果如图 5 所示. 其中，B3C_bi 由双通道编码器和相应的解码器组成，不包含任何聚类优化模块. B3C_sc 在 B3C_bi 的基础上结合了自训练聚类模块，而 B3C_tc 则在 B3C_sc 的基础上增加了三元中心损失. 最后，B3C 为包含协同聚类模块的完整模型.

从实验结果可知：(a)B3C_sc 性能优于 B3C_bi，特别是在无监督聚类指标 NMI 和 ARI 上. 具体来说，与 B3C_bi 相比，B3C_sc 在 ACC 和 F1 指标上具有平均 1.62%和 1.89%和性能提升，而在 NMI 和 ARI 指标上更是有 5.32%和 3.84%的性能提升. 这证

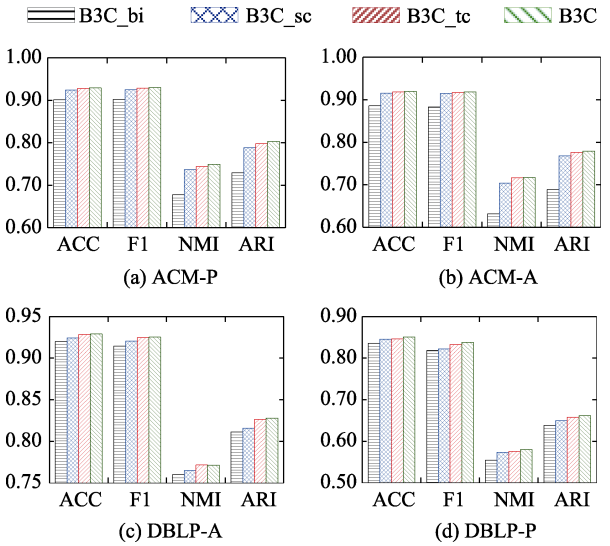


图 5　消融实验

明了学习任务驱动的节点表示的有效性(即联合自训练聚类和表示学习)；(b)由于聚类过程中，B3C_tc 能够进一步缩小簇内节点距离，并扩大簇间节点距离，其性能相较于 B3C_sc 更优；(c)B3C 在聚类优化模块的基础上综合考虑了异质节点间的相关性，因此具有最优的实验结果.

5.2.4　参数分析

本小节评估平衡系数对损失函数的影响. 实验每次改变一个系数并将其余系数设为默认值. λ 和 β 变化区间为 $[10^{-2},10^{2}]$，γ 变化区间为 $[10^{-3},10]$.

λ、β、γ 的默认值分别设为 10、1、0.1. 实验结果如图 6 所示. ACC 和 NMI 随着系数的增大先增大后减小. 这是因为随着系数的增大，模型更加关注聚类优化策略. 系数过大时，生成的表示会偏向于特定损失，不再具有代表性. 总体上，当参数在较大范围内变化时，本文提出的方法性能相对稳定，验证了方法的健壮性.

5.2.5　可视化分析

为了提供更加直观的理解，本文对 B3C 和所有基准方法进行可视化分析. 由于篇幅有限，本文仅展示 ACM-P 上的实验结果. 实验使用 t-SNE[48]在 2 维平面绘制不同方法得到的节点表示，结果如图 7 所示，其中不同颜色的点表示不同的簇. B3C_bi 移除了 B3C 自训练聚类模块和三元中心损失，B3C_sc 在 B3C_bi 的基础上加上了自训练模块，B3C_tc 则在 B3C_sc 上添加了三元中心损失，B3C 为本文提出的完整算法.

可以看出，学习聚类导向表示的方法(DAEGC、AGE、O2MA、MCGC、B3C_sc、B3C_tc 和 B3C)很好地分离了不同的簇，验证了学习任务导向节点

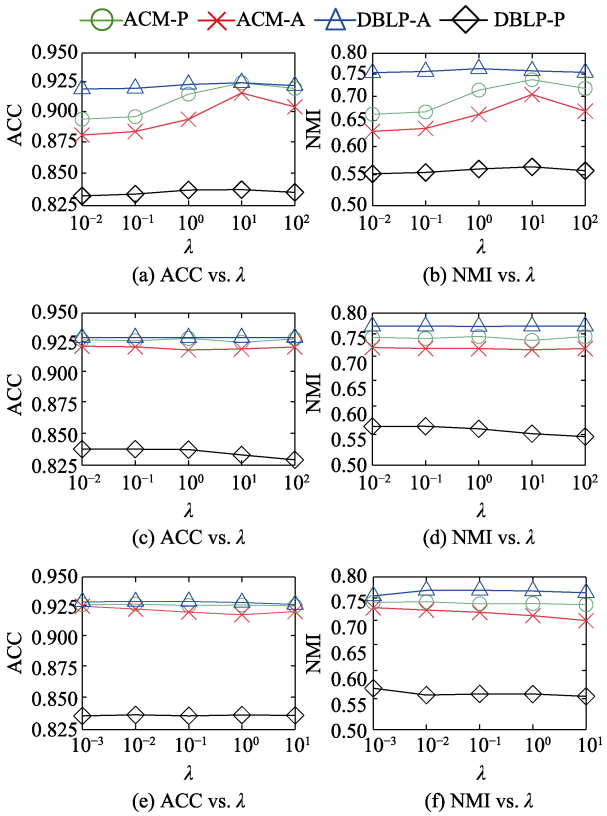


图 6　参数分析

表示的有效性. 尽管 DMGI 没有针对聚类进行优化，但由于能捕获网络的全局特性，因此具有较好的可视化结果. 相比于 AGE、O2MA 和 MCGC，B3C_bi 和 B3C_sc 簇内节点更加紧密，而 B3C_sc 不同簇之间的间隔更加清晰，这是因为 B3C_sc 在 B3C_bi 的基础上应用了自训练聚类. 然而，如图 7(j)所示，不同簇之间仍然彼此靠近，并且在边界有少量重叠. 可以利用 B3C_tc 学习更加具有区分度的节点表示，以获得如图 7(k)所示分离度更大的聚类簇. 最后，如图 7(l)所示，B3C 通过捕获异质节点之间的相关性获得了最佳的可视化效果.

6　总　结

本文提出了一个异质信息网络协同聚类算法 B3C. 为了有效地从属性空间和结构空间聚合信息，本文提出了双通道编码器. 为了学习任务驱动的节点表示，B3C 结合了若干用于聚类的优化策略. B3C 首先设计了自训练聚类模块以联合优化节点表示和聚类. 接着，本文采用三元中心损失以凝聚簇内节点和分裂簇间节点，从而学习具有区分度的节点表示. 最后，本文对异质节点进行协同聚类来进一步提升聚类质量. 在真实数据集上进行了全面的实验，实验结果验证了 B3C 的有效性. 此外，扩展 B3C

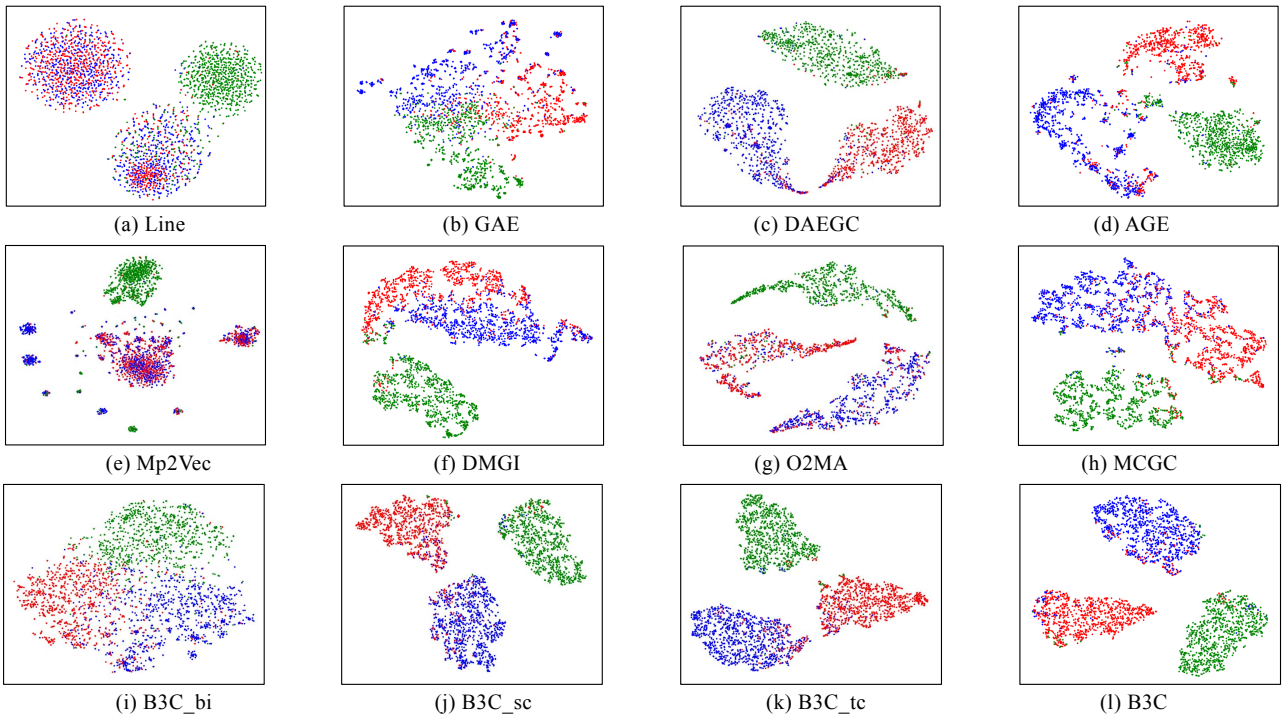


图 7 可视化分析

以对所有类型节点进行聚类也是一个值得研究的问题，后续工作将对其进行深入探讨.

致 谢 感谢国家自然科学基金青年项目（No. 62102351）和浙江省自然科学基金探索青年项目（No. LQ22F020018）的资助；感谢审稿专家和编辑在百忙之中审阅本文!

参 考 文 献

[1] Schaeffer S E. Graph clustering. Computer Science Review, 2007, 1(1): 27-64

[2] Zhao Wei-Ji, Zhang Feng-Bin, Liu Jing-Lian. Review on community detection in complex networks. Computer Science, 2020, 47(2): 10-20 (in Chinese)
(赵卫绩, 张凤斌, 刘井莲. 复杂网络社区发现研究进展. 计算机科学, 2020, 47(2): 10-20)

[3] Yang Bo, Liu Da-You, Liu Ji-Ming, et al. Complex network clustering algorithms. Journal of Software, 2009, 20(1): 54-66 (in Chinese)
(杨博, 刘大有, 金弟, 等. 复杂网络聚类方法. 软件学报, 2009, 20(1): 54-66)

[4] Tian Ye, Liu Da-You, Yang Bo. Application of complex networks clustering algorithm in biological networks. Journal of Frontiers of Computer Science and Technology, 2010, 4(4): 330-337 (in Chinese)
(田野, 刘大有, 杨博. 复杂网络聚类算法在生物网络中的应用. 计算机科学与探索, 2010, 4(4): 330)

[5] Zhao Chuan, Zhang Kai-Han, Liang Ji-ye. Asymmetric recommendation algorithm in heterogeneous information network. Journal of Frontiers of Computer Science and Technology, 2020, 14(6): 939-946 (in Chinese)
(赵传, 张凯涵, 梁吉业. 非对称的异质信息网络推荐算法. 计算机科学与探索, 2020, 14(6): 939-946)

[6] Tang J, Qu M, Wang M, et al. Line: large-scale information network embedding//Proceedings of the International Conference on World Wide Web. Florence, Italy, 2015: 1067-1077

[7] Cui G, Zhou J, Yang C, et al. Adaptive graph encoder for attributed graph embedding//Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining. 2020: 976-985

[8] Bo D, Wang X, Shi C, et al. Structural deep clustering network//Proceedings of the Web Conference. Taipei, China, 2020: 1400-1410

[9] Shi Chuan, Wang Rui-Jia, Wang Xiao. Survey on heterogeneous information networks analysis and applications. Journal of Software, 2021, 33(2): 598-621 (in Chinese)
(石川, 王睿嘉, 王啸. 异质信息网络分析与应用综述. 软件学报, 2021, 33(2): 598-621)

[10] Liu Jia-Wei, Shi Chuan, Yang Cheng, et al. Heterogeneous information network based recommender systems: A survey. Journal of Cyber Security, 2021, 6(5): 1-16 (in Chinese)
(刘佳玮, 石川, 杨成, 等. 基于异质信息网络的推荐系统研究综述. 信息安全学报, 2021, 6(5): 1-16)

[11] Wang X, Ji H, Shi C, et al. Heterogeneous graph attention network//Proceedings of the Web Conference. San Francisco, USA, 2019: 2022-2032

[12] Zhang C, Song D, Huang C, et al. Heterogeneous graph neural network//Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining. Anchorage, USA, 2019: 793-803

[13] Dong Y, Chawla N V, Swami A. Metapath2vec: Scalable representation learning for heterogeneous networks//Proceedings of the ACM International Conference on Knowledge Discovery and

Data Mining. Halifax, Canada, 2017: 135-144
[14] Park C, Kim D, Han J, et al. Unsupervised attributed multiplex network embedding//Proceedings of the AAAI Conference on Artificial Intelligence. New York, USA, 2020: 5371-5378
[15] Fan S, Wang X, Shi C, et al. One2multi graph autoencoder for multi-view graph clustering//Proceedings of The Web Conference. Taipei, China, 2020: 3070-3076
[16] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks//Proceedings of the International Conference on Learning Representations. Toulon, France, 2017
[17] Wang X, Zhu M, Bo D, et al. Am-gcn: adaptive multi-channel graph convolutional networks//Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining. 2020: 1243-1253
[18] Wu F, Souza A, Zhang T, et al. Simplifying graph convolutional networks//Proceedings of the ACM International Conference on Machine Learning. Long Beach, USA, 2019: 6861-6871
[19] Li X, Wu Y, Ester M, et al. Semi-supervised clustering in attributed heterogeneous information networks//Proceedings of the International Conference on World Wide Web. Perth, Australia, 2017: 1621-1629
[20] Lin Z, Kang Z, Zhang L, et al. Multi-view attributed graph clustering. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(2): 1872-1880
[21] Lin Z, Kang Z. Graph filter-based multi-view attributed graph clustering//Proceedings of the International Joint Conference on Artificial Intelligence. Montreal, Canada, 2021: 2723-2729
[22] Pan E, Kang Z. Multi-view contrastive graph clustering//Proceedings of the International Conference on Neural Information Processing Systems. 2021: 2148-2159
[23] Dhillon I S. Co-clustering documents and words using bipartite spectral graph partitioning//Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining. San Francisco, USA, 2001: 269-274
[24] Ding C, Li T, Peng W, et al. Orthogonal nonnegative matrix t-factorizations for clustering//Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining. Philadelphia, USA, 2006: 126-135
[25] Gu Q, Zhou J. Co-clustering on manifolds//Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining. Paris, France, 2009: 359-368
[26] Shang F, Jiao L, Wang F. Graph dual regularization non-negative matrix factorization for co-clustering. Pattern Recognition, 2012, 45(6): 2237-2250
[27] Nie F, Wang X, Deng C, et al. Learning a structured optimal bipartite graph for co-clustering//Proceedings of the International Conference on Neural Information Processing Systems. Long Beach, USA, 2017: 4132-4141
[28] Zhang X, Li H, Liang W, et al. Multi-type co-clustering of general heterogeneous information networks via nonnegative matrix tri-factorization//Proceedings of the IEEE International Conference on Data Mining. Barcelona, Spain, 2016: 1353-1358
[29] Ji Y, Shi C, Fang Y, et al. Semi-supervised co-clustering on attributed heterogeneous information networks. Information Processing and Management, 2020, 57(6): 102338
[30] He X, Zhou Y, Zhou Z, et al. Triplet-center loss for multi-view 3d object retrieval//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA, 2018: 1945-1954
[31] Clauset A, Newman M E J, Moore C. Finding community structure in very large networks. Physical Review E, 2004, 70(6): 066111
[32] Newman M E J. Modularity and community structure in networks. Proceedings of the National Academy of Sciences, 2006, 103(23): 8577-8582
[33] Blondel V D, Guillaume J L, Lambiotte R, et al. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008, 2008(10): P10008
[34] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: online learning of social representations//Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining. New York, USA, 2014: 701-710
[35] Rong X. Word2vec parameter learning explained. arXiv preprint arXiv:1411.2738, 2014
[36] Grover A, Leskovec J. Node2vec: scalable feature learning for networks//Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining. San Francisco, USA, 2016: 855-864
[37] Hartigan J A, Wong M A. Algorithm AS 136: a *k*-means clustering algorithm. Journal of the Royal Statistical Society, Series C (Applied Statistics), 1979, 28(1): 100-108
[38] Wang X, Cui P, Wang J, et al. Community preserving network embedding//Proceedings of the AAAI Conference on Artificial Intelligence. San Francisco, USA, 2017: 203-209
[39] Cavallari S, Zheng V W, Cai H, et al. Learning community embedding with community detection and node embedding on graphs//Proceedings of the ACM Conference on Information and Knowledge Management. Singapore, 2017: 377-386
[40] Li Y, Wang Y, Zhang T, et al. Learning network embedding with community structural information//Proceedings of the International Joint Conference on Artificial Intelligence. Macao, China, 2019: 2937-2943
[41] Zhang Y, Lyu T, Zhang Y. Cosine: community-preserving social network embedding from information diffusion cascades//Proceedings of the AAAI Conference on Artificial Intelligence. New Orleans, USA, 2018: 2620-2627
[42] Kipf T N, Welling M. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308, 2016
[43] Cao S, Lu W, Xu Q. Deep neural networks for learning graph representations//Proceedings of the AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 1145-1152
[44] Pan S, Hu R, Fung S, et al. Learning graph embedding with adversarial training methods. IEEE Transactions on Cybernetics, 2019, 50(6): 2475-2487
[45] Xie J, Girshick R, Farhadi A. Unsupervised deep embedding for clustering analysis//Proceedings of the International Conference on Machine Learning. New York, USA, 2016: 478-487
[46] Guo X, Gao L, Liu X, et al. Improved deep embedded clustering with local structure preservation//Proceedings of the International Joint Conference on Artificial Intelligence. Melbourne, Australia, 2017: 1753-1759
[47] Wang C, Pan S, Hu R, et al. Attributed graph clustering: a deep attentional embedding approach//Proceedings of the International Joint Conference on Artificial Intelligence. Macao, China, 2019: 3670-3676
[48] Veličković P, Fedus W, Hamilton W L, et al. Deep graph infomax//Proceedings of the International Conference for Learning Representations. New Orleans, USA, 2019
[49] Zeyer A, Merboldt A, Schlüter R, et al. A comprehensive analysis

on attention models//Proceedings of the International Conference on Neural Information Processing Systems: Workshop IRASL. Montréal, Canada, 2018

[50] Knyazev B, Taylor G W, Amer M. Understanding attention and generalization in graph neural networks//Proceedings of the International Conference on Neural Information Processing Systems. Vancouver, Canada, 2019: 4202-4212

[51] Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs//Proceedings of the International Conference on Neural Information Processing Systems. Long Beach, USA, 2017: 1025-1035

[52] Xu K, Li C, Tian Y, et al. Representation learning on graphs with jumping knowledge networks//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018: 5453-5462

[53] Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of Machine Learning Research, 2008, 9(11): 2579-2605

[54] Yun S, Jeong M, Kim R, et al. Graph transformer networks//Proceedings of the International Conference on Neural Information Processing Systems. Vancouver, Canada, 2019: 11983-11993

[55] Fu X, Zhang J, Meng Z, et al. Magnn: metapath aggregated graph neural network for heterogeneous graph embedding//Proceedings of the Web Conference. Taipei, China, 2020: 2331-2341

[56] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. Journal of Machine Learning Research, 2010, 9: 249-256

[57] Kingma D P, Ba J. Adam: a method for stochastic optimization//Proceedings of the International Conference for Learning Representations. San Diego, USA, 2015

QIU Lin-Shan, Ph.D. candidate. His research interests include graph data processing and graph representation learning.

FANG Zi-Quan, Ph.D., professor. His research interests include spatial-temporal data management and analytics.

CHEN Lu, Ph.D., professor. Her research interests include database and big data management and analytics.

ZHANG Tian-Ming, Ph.D., distinguished research fellow. Her research interests include graph data management and deep learning.

LI Tian-Yi, Ph.D., assistant professor. Her research interests include spatial-temporal data management and analytics.

Background

Network clustering is a fundamental task for data mining, which aims to partition nodes of a network into several disjointed groups, such that nodes in a group are similar to each other. Network clustering has drawn increasing attention due to its omnipresent real-world applications, including community detection, social recommendation, abnormal detection, etc. However, most of the existing network-oriented clustering studies are designed for homogeneous networks, where all the nodes are of the same type. Whereas, networks in real life essentially consist of objects and interactions of various types. For instance, social network is composed of users, videos, images and so on. Friendship could be established between users. Besides, users can post a video or like some images. Similarly, online shopping network comprises sellers, buyers, commodities, and so on. The interactions include purchase, favorite, like, etc. These complex networks can be modeled as heterogeneous information networks (HINs), which provide complete structure and rich semantic information for data mining. The ability of HINs to model complex relations between multiple-typed objects has motivated increasing studies for HIN-based data mining, among which HIN clustering is a fundamental task. However, exiting HIN clustering studies either ignore the node features and simply merge node features with structural information without considering the weights of HIN nodes and HIN structures to HIN clustering. Moreover, they focus on clustering target nodes of a single type while neglecting the adjuvant effects that other nodes of other types in HINs may bring. Last but not least, they cannot learn discriminative representations that enable tightening similar nodes while separating dissimilar nodes.

Motivated by the mentioned issues above, this paper proposes Bi-Channel Co-Clustering algorithm for heterogeneous information networks, i.e., B3C. which is capable of fully capturing the hidden correlations between multi-typed nodes while clustering multiple types of nodes simultaneously to improve clustering. To effectively aggregate information from the structure space and the feature space, this paper proposes a bi-channel encoder, which serves as the backbone of the B3C algorithm. For the sake of learning a task-friendly representation, B3C is equipped with several optimization strategies derived for clustering. B3C first employs the self-training based clustering module to optimize the representation and the clustering jointly. Then, B3C adopts the triplet-center loss to facilitate a discriminative embedding by tightening the intra-class objects while separating the inter-class ones. Finally, B3C leverages the learned representation for co-clustering heterogeneous nodes. Extensive experiments on real-world datasets illustrate that B3C significantly outperforms state-of-the-art methods.

This work was supported in part by the Young Scientists Fund of the National Natural Science Foundation of China under Grant No. 62102351, and the Natural Science Foundation of Zhejiang Province of China under Grant No.LQ22F020018.