计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210300060

基于矩阵分解的属性网络嵌入和社区发现算法

徐新黎　肖云月　龙海霞　杨旭华　毛剑飞
浙江工业大学计算机科学与技术学院　杭州 310023
(xxl@zjut.edu.cn)

摘　要　属性网络不但包含节点之间复杂的拓扑结构,还包含拥有丰富属性信息的节点,其可以比传统网络更有效地建模现代信息系统,属性网络的社区划分对于分析复杂系统的层次结构、控制信息在网络中的传播和预测网络用户的群体行为等方面具有重要的研究价值。为了更好地利用拓扑结构信息和属性信息进行社区发现,提出了一种基于矩阵分解的属性网络嵌入和社区发现算法(CDEMF)。首先提出基于矩阵分解的属性网络嵌入方法,基于网络局部链接信息计算相邻节点的相似性,将其与属性接近度联合建模,通过矩阵分解的分布式算法得到每个节点对应的低维嵌入向量,即把网络节点映射为低维向量表示的数据点集合。接着提出基于曲率和模块度的社区划分方法,自动确定数据点集合中蕴含的社区数量,并通过对数据点集合聚类完成属性网络社区划分。在真实网络数据集上,将CDEMF方法与其他8种知名算法进行比较,实验结果表明CDEMF具有良好的性能。

关键词:属性网络嵌入;矩阵分解;自动聚类;社区发现;曲率
中图法分类号　TP391

Attributed Network Embedding Based on Matrix Factorization and Community Detection

XU Xin-li,XIAO Yun-yue,LONG Hai-xia,YANG Xu-hua and MAO Jian-fei
College of Computer Science and Technology,Zhejiang University of Technology,Hangzhou 310023,China

Abstract　An attributed network contains not only the complex topological structure but also the nodes with rich attribute information. It can be used to more effectively model modern information systems than traditional networks. Community detection of the attributed network has important research value in hierarchical analysis of complex systems,control of information propagation in the network,and prediction of group behavior of network users. In order to make better use of topology information and attribute information for community discovery,an attributed network embedding based on matrix factorization and community detection(CDEMF) are proposed. First,an attributed network embedding method based on matrix factorization is proposed to model the attributed proximity and the similarity of adjacent nodes calculated in term of the local link information of the network,where the low-dimensional embedding vector corresponding to each node can be obtained by a distributed algorithm of matrix decomposition,that is,the network nodes can be mapped into a collection of data points represented by low-dimensional vectors. Then the community detection method based on curvature and modularity is developed to achieve attributed network community division by clustering the data point set,which can automatically determine the number of communities contained in the data point set. CDEMF is compared with the other 8 kinds of well-known approaches on public real network datasets. The experimental results demonstrate the effectiveness and superiority of CDEMF.

Keywords　Attributed network embedding,Matrix factorization,Automatic clustering,Community detection,Curvature

1　引言

随着信息网络的快速发展,现实世界中出现了越来越多的属性网络,如社交网络中的微博网络、蛋白质网络、文献引文网络等。与单纯由点和边组成的纯拓扑结构网络不同的是,属性网络中的每个节点还拥有丰富的属性信息,如引文网络中对论文的描述、微博社交网络中对博客的描述等。近年来,怎样利用属性网络的拓扑信息和节点属性信息实现社区挖掘越来越受到学者们的关注。

近十年已经出现了不少社区发现的方法,如Fast Unfolding算法[1]、标签传播算法[2]、GEMSEC(Graph Embedding with Self Clustering)算法[3]、vGRAPH(Generative Model for Joint Community Detection and Node Representation Learning)算法[4]等,这些方法通常基于拓扑信息寻找网络中拓扑

到稿日期:2021-03-05　返修日期:2021-06-08　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金项目(61773348);浙江省公益科技计划项目(LGG20F020017);浙江省自然科学基金项目(LQ18F030015)
This work was supported by the National Natural Science Foundation of China(61773348),Zhejiang Public Welfare Science and Technology Plan Project(LGG20F020017) and Natural Science Foundation of Zhejiang Province(LQ18F030015).
通信作者:毛剑飞(mjf@zjut.edu.cn)

结构连接紧密的部分。一般认为，网络拓扑对社区的形成有很大的作用，但对于现实世界中的网络，我们发现节点的属性信息对社区的形成同样有不可忽视的作用。一旦忽略这些丰富的属性信息对社区的影响，很可能会导致社区的错误划分。

综合利用拓扑信息和属性信息进行社区发现可以有效避免单一利用拓扑结构信息划分社区的错误。目前，越来越多的研究者提出了将节点属性信息纳入考量的算法。Yang 等[5]提出联合边结构和节点属性的社区发现算法，通过对网络拓扑结构和节点属性之间的相互关系建模，并且假设每个节点 u 对社区 c 都有一个非负相关权重 F_{uc}，当 F_{uc} 大于一定的阈值时，节点 u 属于社区 c，该算法可以检测到重叠社区；Jin 等[6]基于非负矩阵分解框架提出一种属性网络社区发现模型，采用一个带先验的转移概率矩阵来描述社区与内容类簇之间的关系。由于属性网络节点的属性信息维度易达到成千上万，因此上述算法复杂度均较高，而且并没有深入挖掘拓扑信息。

属性网络嵌入可以将属性网络中各个高维的节点嵌入到向量空间中，并表示为一个低维向量。该低维向量可以看作节点的特征向量，并且可以同时保持节点在网络拓扑结构和属性信息的特征[7]。Huang 等[8]提出一种属性网络嵌入框架，将拓扑信息和属性信息联合表征为嵌入向量，并且提出了分布式算法以加快矩阵分解运算。Li 等[9]提出 NetFS(Robust Unsupervised Feature Selection Framework)框架，其利用拓扑信息选择出内容信息相关特征的子集，将潜在表征向量嵌入到特征选择中，再运用 k-means 算法进行聚类。NetFS 利用内容信息有效减少了噪声链接对潜在表征的负面影响。上述算法在得到节点低维向量表示后，可以利用 k-means 算法实现社区划分，但需要人为指定社区个数，另外这些算法往往只利用邻接矩阵来挖掘拓扑信息，很可能会造成信息的浪费和错误的划分。

为了在属性网络中充分利用拓扑信息和属性信息，提高属性网络社区发现的准确性，并且自动确定社区个数，本文提出了基于矩阵分解的属性网络嵌入和社区发现方法 CDEMF(Attributed Network Embedding Based on Matrix Factorization and Community Detection)。CDEMF 首先深入挖掘网络的拓扑信息，将节点的共同邻居信息和属性信息联合建模，并通过分布式的矩阵分解算法计算得到联合嵌入向量；然后用提出的基于曲率和模块度的社区划分方法来自动确定社区个数，并且得到社区划分的结果。实验结果表明，CDEMF 具有良好的属性网络社区划分性能。

本文第 2 节介绍了相关工作；第 3 节分析了基于矩阵分解的属性网络嵌入和社区发现算法；第 4 节进行了数值仿真和结果分析；最后总结全文并展望未来工作。

2 相关工作

2.1 网络嵌入

为了解决网络数据量大和维度高的问题，学者们提出了一些网络嵌入的方法。例如，Perozzi 等[10]将自然语言处理的思想用在网络嵌入上，提出了网络嵌入方法 DeepWalk，其将每个节点当作单词，将随机游走后得到的序列当作句子来学习潜在的表征。Tang 等[11]提出另一种网络嵌入方法 LINE(Large-scale Information Network Embedding)，适用于任何类型的网络，该方法优化了目标函数，还利用边缘采样算法克服了随机梯度下降的局限性。DeepWalk 和 LINE 虽然降低了网络嵌入的难度，但它们仅考虑网络拓扑结构信息，忽略了大量的节点属性信息。

Wang 等[12]提出的基于多视图的聚类算法 GMC(Graph-based Multi-view Clustering)，把每个视图融合在一起以构建统一的图矩阵，可以直接给出聚类结果。Sun 等[13]提出的用于属性网络的嵌入方法 NEC(Network Embedding for Community Detection in Attributed Networks)，首先通过图卷积自编码器对拓扑信息和属性信息进行嵌入，然后用梯度下降算法进行优化，学习节点嵌入。Ye 等[14]将结构视图、边权视图、属性视图进行融合，提出了一种多视图的网络嵌入算法 MVENR(Network Representation Learning Algorithm based on Multi-view Integration)，该算法弥补了信息单一的不足。Zhang 等[15]提出的 GUCD(Community-centric Graph Convolutional Network for Unsupervised Community Detection)算法，基于图卷积神经网和自编码器，引入了以社区为中心的解码器，并以无监督的方式分别重构网络结构和节点属性。但是 GMC 算法、MVENR 算法主要针对多视图数据，对单一视图的网络社区划分的效果并不理想；NEC 和 GUCD 算法虽然考虑了拓扑和属性两方面的信息，但是对拓扑信息的利用较单一，只利用了邻接矩阵，并没有挖掘更多的信息。

2.2 聚类算法

近年来，学者们提出了一些经典的聚类算法。例如，Ester 等[16]提出的 DBSCAN(Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise)算法，把簇定义为密度相连的点的最大集合，把高密度的区域划分成簇，该算法不需要提前给出簇的个数。Rodriguez 等[17]提出了基于密度的聚类算法，定义了局部密度指标和最小距离指标，并通过决策图手工确定聚类中心。Chen 等[18]提出一种基于启发式方法确定簇个数的谱聚类算法，该算法首先根据数据的密度分布选择簇个数和类中心点，然后用一种改进的 k-means 算法进行聚类。

一些学者提出的算法可以在多次运行 k-means 算法的情况下确定正确的簇个数。如 Tibshirani 等[19]提出的 gap 方法定义了间隔统计量(gap statistic)，用间隔统计量来确定类中簇的个数，该算法的计算量较大。Sugar 等[20]提出的 jump 方法，首先运行不同聚类数目(k)的 k-means 算法，同时计算相应的失真(每个数据点与其最近的簇中心之间的平均距离)，然后计算转换后失真的最大“jump”值，其对应的 k 值即为最后确定的簇个数。Zhang 等[21]提出的 Curvature 方法定义了曲率指标，计算不同 k 值下的曲率指标，取最大值对应的 k 值即为簇的个数。这些算法虽然能确定社区的个数，但是正确率较低。

3 基于矩阵分解的属性网络嵌入和社区发现方法

3.1 问题描述

一个既包含链接信息又包含属性信息的复杂网络可以表示为 $G=(V,E,\boldsymbol{F})$，其中 V 为节点集合；E 为链接边集合；所有节点的属性可以表示为一个 $n\times m$ 的属性信息矩阵 $\boldsymbol{F}$，n 是

网络的节点个数，m 是节点属性类别个数。对于属性网络，图 G 可以建模为无向无权图，用 e_{ij} 表示节点 i 和节点 j 之间的连接关系，若节点 i 和节点 j 之间存在边，那么 $e_{ij}=1$，否则 $e_{ij}=0$。所有节点的连边关系构成一个邻接矩阵 $\boldsymbol{A}$。本文的主要符号及定义如表 1 所列。

表 1 主要符号及定义

Table 1 Main symbols and definitions

Symbols	Definitions
$n=\|V\|$	Number of nodes
m	Number of node attribute categories
d	Embedding vector dimension
$\boldsymbol{A}\in\boldsymbol{R}^{n\times n}$	Adjacency matrix
$\boldsymbol{F}\in\boldsymbol{R}^{n\times m}$	Attribute information matrix
$\boldsymbol{SA}\in\boldsymbol{R}^{n\times n}$	Node similarity matrix
$\boldsymbol{SF}\in\boldsymbol{R}^{n\times n}$	Attribute proximity matrix
$\boldsymbol{H}\in\boldsymbol{R}^{n\times d}$	Embedding vector

本文社区发现问题可以描述为：给定一个由 n 个节点组成的网络 G，每个节点有 m 个属性，给予节点属性矩阵 $\boldsymbol{F}$ 和邻接矩阵 $\boldsymbol{A}$ 后进行社区发现。首先计算节点相似度矩阵 $\boldsymbol{SA}$ 和属性接近度矩阵 $\boldsymbol{SF}$，基于矩阵分解进行联合建模，计算得到每个节点的 d 维嵌入向量；再对节点的嵌入向量 $\boldsymbol{H}$ 进行聚类实现社区发现，即标记节点的社区类别标签，使得社区内节点密集相连并且属性相似，不同社区的节点连接稀疏或者属性不相似。

3.2 基于矩阵分解的属性网络嵌入

针对一些社区发现算法[7-8,22]没有充分结合社区拓扑结构信息和属性信息的问题，本文提出了一种基于矩阵分解的属性网络嵌入框架。该框架根据邻接矩阵计算相似度，基于矩阵分解将属性信息和拓扑信息联合表征，然后通过分布式算法计算得到联合嵌入向量 $\boldsymbol{H}$。

3.2.1 融合邻居信息的网络拓扑建模

为了使联合嵌入向量 $\boldsymbol{H}$ 能更好地表示属性网络，必须尽可能保持网络拓扑的节点特性。一般认为，在网络拓扑结构 G 中，通常拥有更多共同邻居的两个节点比拥有共同邻居较少的两个节点更为相似。因此，可以采用含有共同邻居信息的节点相似度矩阵 $\boldsymbol{SA}$ 来约束节点向量表示，由此提出损失函数来最小化两个相邻节点的嵌入向量的差异。

$$J_G=\sum_{(i,j)\in E}\boldsymbol{SA}_{ij}\,\|\boldsymbol{h}_i-\boldsymbol{h}_j\|_2 \tag{1}$$

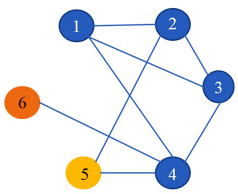
其中，$\boldsymbol{h}_i$，$\boldsymbol{h}_j$ 分别代表节点 i 和节点 j 的嵌入向量，$\boldsymbol{SA}_{ij}$ 为节点 i 和节点 j 的相似性。式(1)的基本思想是最小化损失函数 J_G，即 $\boldsymbol{SA}_{ij}$ 越大，两个节点的共同邻居数越多，$\boldsymbol{h}_i$ 和 $\boldsymbol{h}_j$ 的差距就越小，表明节点 i 和节点 j 越相似。这里采用 L2 范数作为嵌入向量之间的差异度量，目的是减轻异常值和缺失数据的负面影响[7]。

$\boldsymbol{SA}_{ij}$ 的定义如下：

$$\boldsymbol{SA}_{ij}=\begin{cases}\dfrac{|c_i\cap c_j|}{n}, & e_{ij}\neq 0\\ 0, & e_{ij}=0\end{cases} \tag{2}$$

其中，$|c_i\cap c_j|$ 为节点 i 和节点 j 的共同邻居个数。在节点 i 与节点 j 相连的情况下，$\boldsymbol{SA}_{ij}$ 为两个节点的共同邻居数在总节点数中的占比；在两节点不相连的情况下，$\boldsymbol{SA}_{ij}$ 为 0。

融合共同邻居信息的节点相似度矩阵 $\boldsymbol{SA}$ 比邻接矩阵 $\boldsymbol{A}$ 更能挖掘出拓扑方面的信息。以节点数为 6 的网络为例，对应的相似度矩阵 $\boldsymbol{SA}$ 和邻接矩阵 $\boldsymbol{A}$ 如图 1 所示。例如，节点 1 的邻居为节点 2、节点 3、节点 4，对应的相似度 $\boldsymbol{SA}$ 分别为 1/6，2/6，1/6，最小化损失函数 J_G 会迫使 $\boldsymbol{h}_1$ 和 $\boldsymbol{h}_3$ 比 $\boldsymbol{h}_1$ 和 $\boldsymbol{h}_2$、$\boldsymbol{h}_1$ 和 $\boldsymbol{h}_4$ 更加相近。从图 1 可以看到，节点 1 在拓扑结构上与节点 3 更加相似。而在邻接矩阵中，节点 1 和节点 2、节点 3、节点 4 的相似程度并没有区别。再以节点 4 为例，它的邻居为节点 1、节点 3、节点 5、节点 6，但是节点 4 与节点 5、节点 6 并没有共同邻居，因此节点 4 与节点 1、节点 3、节点 5、节点 6 之间的相似度分别为 1/6，1/6，0 和 0，即节点 4 与节点 1、节点 3 会更加相似，而非节点 5、节点 6。



Network

(a)节点数为 6 的网络

	n_1	n_2	n_3	n_4	n_5	n_6
n_1	0	1/6	2/6	1/6	0	0
n_2	1/6	0	1/6	0	0	0
n_3	2/6	1/6	0	1/6	0	0
n_4	1/6	0	1/6	0	0	0
n_5	0	0	0	0	0	0
n_6	0	0	0	0	0	0

Similarity matrix

(b)网络的相似度矩阵 $\boldsymbol{SA}$

	n_1	n_2	n_3	n_4	n_5	n_6
n_1	0	1	1	1	0	0
n_2	1	0	1	0	1	0
n_3	1	1	0	1	0	0
n_4	1	0	1	0	1	1
n_5	0	1	0	1	0	0
n_6	0	0	0	1	0	0

Topology matrix

(c)邻接矩阵 $\boldsymbol{A}$

图 1 节点相似度分析

Fig. 1 Node similarity analysis

3.2.2 基于矩阵分解的联合嵌入

为了使属性网络的节点嵌入表征 $\boldsymbol{H}$ 可以很好地同时保持节点属性和网络拓扑的接近度，本文对网络拓扑和属性信息联合建模[8]，目标函数如下：

$$\min_{\boldsymbol{H}} J=J_A+\lambda J_G \tag{3}$$

其中，λ 作为一个标量，是拓扑信息与属性信息所占比例的权衡，当 $\lambda=0$ 时，表示嵌入模型只考虑了属性信息；当 λ 较大时，表示嵌入模型更多地考虑了拓扑信息。

属性信息的损失函数 J_A 可以表示为[7]：

$$J_A=\|\boldsymbol{SF}-\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}\|_F^2=\sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{SF}_{ij}-\boldsymbol{h}_i\boldsymbol{h}_j^{\mathrm{T}})^2 \tag{4}$$

其中，$\boldsymbol{SF}$ 是非负矩阵，$\|\cdot\|_F^2$ 表示 Frobenius 范式。为了使 $\boldsymbol{H}$ 更好地保持属性接近度，根据对称矩阵分解[23]，可以将 $\boldsymbol{SF}$ 分解为两个矩阵 $\boldsymbol{H}\in\boldsymbol{R}^{n\times d}$ 和 $\boldsymbol{H}^{\mathrm{T}}\in\boldsymbol{R}^{d\times n}$ 的积，即 $\boldsymbol{SF}\approx\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$。矩阵 $\boldsymbol{SF}$ 中的每个向量 $\boldsymbol{SF}_{ij}$ 都可以通过 $\boldsymbol{H}_i$ 与 $\boldsymbol{H}_j^{\mathrm{T}}$ 的点积来表示，同时用基于欧氏距离的目标函数来逼近 $\boldsymbol{SF}$ 和 $\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$ 之间的误差，只有当 $\boldsymbol{SF}=\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$ 时，式(4)才能得到最小值，也就是 0。

通过最小化该损失函数，求解矩阵分解因式，即用 $\boldsymbol{H}$ 和

H^{T}的积来近似逼近属性接近度矩阵，可以得到嵌入表征向量$\boldsymbol{H}$。这里的属性接近度矩阵$\boldsymbol{SF}$由相似度度量标准计算得到，可以采用余弦相似度，计算公式如下：

$$\boldsymbol{SF}_{ij}=\frac{\langle \boldsymbol{F}_i,\boldsymbol{F}_j\rangle}{\|\boldsymbol{F}_i\|\times\|\boldsymbol{F}_j\|} \tag{5}$$

其中，$\langle\cdot,\cdot\rangle$为内积算子，$\|\cdot\|$表示欧氏范数，$\boldsymbol{F}_i$表示节点$i$的属性信息。

因此，得到最终的联合优化目标函数为：

$$\min_{\boldsymbol{H}} J=\sum_{i=1}^{n}\sum_{j=1}^{n}(\boldsymbol{SF}_{ij}-\boldsymbol{h}_i\boldsymbol{h}_j^{\mathrm{T}})^2+\lambda\sum_{(i,j)\in E}\boldsymbol{SA}_{ij}\|\boldsymbol{h}_i-\boldsymbol{h}_j\|_2 \tag{6}$$

本文采用分布式算法 ADMM[8]对联合嵌入模型进行求解得到嵌入向量$\boldsymbol{H}$。ADMM 将J重新表述为一个双凸优化问题，然后用含有t个线程的分布式算法对其进行求解。

首先令$\boldsymbol{Z}=\boldsymbol{H}$，将式(6)重新表示为如下线性约束问题：

$$\min_{\boldsymbol{H}}\sum_{i=1}^{n}\|\boldsymbol{SF}_i-\boldsymbol{h}_i\boldsymbol{Z}^{\mathrm{T}}\|_2^2+\lambda\sum_{(i,j)\in E}\boldsymbol{SA}_{ij}\|\boldsymbol{h}_i-\boldsymbol{z}_j\|_2 \tag{7}$$

$$\text{s.t. } \boldsymbol{h}_i=\boldsymbol{z}_i,i=1,\cdots,n$$

可以得到该凸优化问题的增广拉格朗日函数为：

$$L=\sum_{i=1}^{n}\|\boldsymbol{SF}_i-\boldsymbol{h}_i\boldsymbol{Z}^{\mathrm{T}}\|_2^2+\lambda\sum_{(i,j)\in E}\boldsymbol{SA}_{ij}\|\boldsymbol{h}_i-\boldsymbol{z}_j\|_2+\frac{\rho}{2}\sum_{i=1}^{n}(\|\boldsymbol{h}_i-\boldsymbol{z}_j+\boldsymbol{\mu}_i\|_2^2-\|\boldsymbol{\mu}_i\|_2^2) \tag{8}$$

其中，ρ是增广拉格朗日参数，且$\rho>0$；u_i是对偶变量，$i=1,2,\cdots,n$，可以用 ADMM 算法求解，迭代方法为：

$$\boldsymbol{h}_i^{p+1}=\arg\min_{\boldsymbol{h}_i}(\|\boldsymbol{SF}_i-h_i\boldsymbol{Z}^{p\mathrm{T}}\|_2^2+\lambda\sum_{j\in N(i)}\boldsymbol{SA}_{ij}\|\boldsymbol{h}_i-\boldsymbol{z}_j^p\|_2+\frac{\rho}{2}\|\boldsymbol{h}_i-\boldsymbol{z}_i^p+\boldsymbol{u}_i^p\|_2^2) \tag{9}$$

$$\boldsymbol{z}_i^{p+1}=\arg\min_{\boldsymbol{z}_i}(\|\boldsymbol{SF}_i^{\mathrm{T}}-\boldsymbol{H}^{p+1}\boldsymbol{z}_i^{\mathrm{T}}\|_2^2+\lambda\sum_{j\in N(i)}\boldsymbol{SA}_{ij}\|\boldsymbol{z}_i-\boldsymbol{h}_j^{p+1}\|_2+\frac{\rho}{2}\|\boldsymbol{z}_i-\boldsymbol{h}_i^{p+1}+\boldsymbol{u}_i^p\|_2^2) \tag{10}$$

$$\boldsymbol{U}^{p+1}=\boldsymbol{U}^p+(\boldsymbol{H}^{p+1}-\boldsymbol{Z}^{p+1}) \tag{11}$$

其中，p为迭代次数，通过对式(9)、式(10)求导，得到$\boldsymbol{h}_i$和$\boldsymbol{z}_i$的更新规则如下：

$$\boldsymbol{h}_i^{p+1}=\frac{2\boldsymbol{SF}_i\boldsymbol{Z}^p+\lambda\sum_{j\in N(i)}\frac{\boldsymbol{SA}_{ij}\boldsymbol{z}_j^p}{\|\boldsymbol{h}_i^p-\boldsymbol{z}_j^p\|_2}+\rho(\boldsymbol{z}_i^p-\boldsymbol{u}_i^p)}{2\boldsymbol{Z}^{p\mathrm{T}}\boldsymbol{Z}^p+(\lambda\sum_{j\in N(i)}\frac{\boldsymbol{SA}_{ij}}{\|\boldsymbol{h}_i^p-\boldsymbol{z}_j^p\|_2}+\rho)\boldsymbol{I}} \tag{12}$$

$$\boldsymbol{z}_i^{p+1}=\frac{2\boldsymbol{SF}_i\boldsymbol{H}^p+\lambda\sum_{j\in N(i)}\frac{\boldsymbol{SA}_{ij}\boldsymbol{h}_j^p}{\|\boldsymbol{z}_i^p-\boldsymbol{h}_j^p\|_2}+\rho(\boldsymbol{h}_i^p-\boldsymbol{u}_i^p)}{2\boldsymbol{H}^{p\mathrm{T}}\boldsymbol{H}^p+(\lambda\sum_{j\in N(i)}\frac{\boldsymbol{SA}_{ij}}{\|\boldsymbol{z}_i^p-\boldsymbol{h}_j^p\|_2}+\rho)\boldsymbol{I}} \tag{13}$$

其中，$\boldsymbol{I}$表示单位矩阵。每次迭代首先求解与$\boldsymbol{h}_i$相关的最小化问题即式(9)，并根据式(12)更新变量$\boldsymbol{h}$；然后求解与$\boldsymbol{z}_i$相关的最小化问题即式(10)，再根据式(13)更新变量$\boldsymbol{z}$；最后更新对偶变量$\boldsymbol{u}$，不断迭代直到收敛。

融合基于矩阵分解的属性网络嵌入如算法 1 所示。

算法 1 融合基于矩阵分解的属性网络嵌入

输入：属性信息矩阵 **F**，邻接矩阵 **A**，嵌入向量维度 d

输出：嵌入向量 **H**

1. 根据式(2)对 **A** 计算节点相似性矩阵 **SA**
2. 根据式(5)对 **F** 计算属性接近度矩阵 **SF**
3. 设置 p=0，取 **SF** 的 2d 列元素作为 $\mathbf{SF}_0$
4. $\mathbf{H}^p$为 $\mathbf{SF}_0$的左奇异向量，$\mathbf{Z}^p=\mathbf{H}^p$，$\mathbf{U}^p=0$
5. repeat
6. 计算 $\mathbf{Z}^{p\mathrm{T}}\mathbf{Z}^p$/*把 n 个任务分配给 t 个线程*/
7. 根据式(12)更新 h_i^{p+1}(i=1,2,…,n)
8. 计算 $\mathbf{H}^{(p+1)\mathrm{T}}\mathbf{H}^{p+1}$/*把 n 个任务分配给 t 个线程*/
9. 根据式(13)更新 z_i^{p+1}(i=1,2,…,n)
10. $\mathbf{U}^{p+1}\leftarrow\mathbf{U}^p+(\mathbf{H}^{p+1}-\mathbf{Z}^{p+1})$
11. k←k+1
12. until 满足收敛条件
13. 返回 **H**

3.3 自动确定社区个数的社区发现算法

根据算法 1 得到节点的嵌入向量$\boldsymbol{H}$后，可以利用k-means 算法来实现社区发现。k-means 算法简单、聚类效果较优且收敛快，但是它需要人为提前给定聚类数，在社区划分个数未知的情况下无法使用该算法。为了克服这个不足，本文提出了一种基于曲率和模块度自动确定社区个数的聚类算法，其可以根据得到的节点联合嵌入向量$\boldsymbol{H}$自动确定社区个数K，并且返回节点标签。

自动确定聚类数的曲率法是在肘部法的基础上发展而来的，即评估图的“肘部”点对应于评估图曲率最大的点，这里的评估图是通过聚类过程中集群内方差与聚类数目绘制而成的[21]。在评估图中，随着聚类数目k值的增大，群内方差呈逐渐减小的趋势，到达正确k值之前，群内方差缩小幅度较大；超过正确的k值后，群内方差程度迅速变小，然后趋于平缓。

群内方差的定义[21]如下：

$$J(k)=\sum_{j=1}^{k}\sum_{x_i\in P_j}\|\boldsymbol{X}_i-\overline{\boldsymbol{X}_j}\|^2 \tag{14}$$

其中，P_j为社区j的节点集合，$\overline{\boldsymbol{X}_j}$为社区$j$的各节点向量平均值。

Zhang 等[21]认为曲率有缩放的缺陷，因此提出可以用曲率指标代替曲率，从而根据曲率指标$r(k)$确定社区个数，其公式如下：

$$K=\arg\max_k r(k) \tag{15}$$

$$r(k)=\frac{det_k}{det_{k+1}} \tag{16}$$

$$det_k=J(k-1)-J(k) \tag{17}$$

从 20 个来自 UCI 的真实数据集的仿真验证结果可看出[21]，曲率法的正确率高于 CH[24]，KL[25]，JUMP[20]等其他 6 种方法。为了使该方法可以更好地应用于节点属性网络社区个数的确定，同时进一步提高所确定社区个数的正确率，我们将模块度也应用于社区个数的确定中。

模块度(modularity)最初是在 2004 年由 Newman[26]提出，2006 年又被重新进行了定义。Newman 认为模块度越高，表明社区划分的效果越好，且Q值的范围在[−0.5,1]之间。此外，若Q值在[0.3,0.7]区间内，则说明聚类的效果很好[27]。模块度Q的定义如下：

$$Q=\frac{1}{2s}\sum_{u=1}^{C}\left[2l_u-\frac{d_u^2}{2s}\right] \tag{18}$$

其中，u为社区序号，C为社区的总个数，l_u是社区C的总边数，d_u是社区u中所有节点的总度数，s为整个网络节点之间的总边数。

基于曲率和模块度的社区划分方法的流程如算法 2 所示。首先确定嵌入向量在不同聚类数目k值下的最小群内方

差。其次根据式(16)计算曲率指标 $r(k)$，并返回曲率指标最大的值对应的 k 值，记为 k_1，以及第二大的值对应的 k 值，记为 k_2；循环 I_1 次后，确定 I_1 次循环中出现次数最多的 k_1 和 k_2，分别记为 k_a，k_b。再分别计算 T_2 次内以 k_a 和 k_b 为划分个数的最大模块度 Q_1 和 Q_2，并根据式(19)选择 Q_1 和 Q_2 中较大值对应的 k 值，将其作为最后社区划分的个数 K，运行社区数为 K 的 k-means 算法，返回 K 和相应的聚类结果。

$$K=\begin{cases}k_a, & Q_1 \geqslant Q_2\\ k_b, & Q_1 < Q_2\end{cases} \tag{19}$$

算法 2 基于曲率和模块度的社区划分方法

输入：嵌入向量 **H**

输出：社区个数 K，节点标签 Label

```
repeat
    for k=2:(k_max+1) do
        for t=1:T do
            运行 k-means，基于式(14)计算群内方差
        end for
        得到 T 次中群内方差最小值记为 J(k)
    end for
    if k > 1
        基于式(17)和式(16)计算曲率指标 r(k)
    end if
    for k=2:k_max do
        基于式(15)确定 k_1，排除 k_1 后基于式(15)确定 k_2
    end for
until i=1:I_1
确定 I_1 次循环中出现次数最多的 k_1 和 k_2，分别记为 k_a，k_b
for it=1:T_2
    分别运行 K=k_a 时和 K=k_b 时的 k-means，计算模块度 Q_1(it)，Q_2(it)
end for
对 Q_1(it)，Q_2(it)分别取最大值记为 Q_1，Q_2，K 为 Q_1 和 Q_2 中较大值对应的社区值
运行聚类数目为 K 的 k-means 算法，得到节点标签 Label
返回 K，Label
```

3.4 算法流程及复杂度分析

CDEMF 算法的整体框架如图 2 所示，该算法由两个阶段组成，第一阶段为融合邻居信息和属性信息的属性网络嵌入。首先，对属性网络进行属性接近度计算和节点相似性计算，得到矩阵 ***SF*** 和 ***SA***，对两个矩阵进行联合建模，再用分布式算法计算得到嵌入向量 ***H***。第二阶段为基于曲率和模块度的社区划分。根据嵌入向量 ***H***，先计算出第一曲率和第二曲率对应的社区个数，再计算出它们的模块度并进行比较，得到最终的社区划分个数，然而利用 k-means 聚类算法实现社区发现。

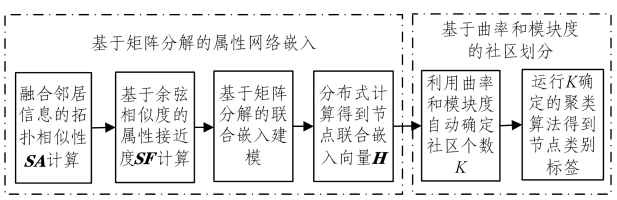


图 2 CDEMF 算法框架图

Fig. 2 CDEMF algorithm framework

CDEMF 的算法 1 中计算相似度的复杂度为 $O(|E|)$，$|E|$ 为边的数量，根据 AANE[7] 可知，算法 1 中分布式计算的复杂度为 $O(nN_F+n^2/t)$，t 为分布式算法的线程数，N_F 为 F 中非零的个数；算法 2 的时间复杂度为 $O(I_1(k_{\max}Tn+k_{\max})+T_2n)$，由于 I_1，$k_{\max}$，T，T_2 远小于 n，因此其复杂度为 $O(I_1Tk_{\max}n)$。CDEMF 算法的总时间复杂度为 $O(|E|+nN_F+n^2/t+I_1Tk_{\max}n)$。

4 数值仿真与结果分析

本节首先给出实验数据集和用来对比的基准算法；然后验证 CDEMF 算法中基于曲率和模块度自动确定社区个数的过程和正确性；其次引入 3 个评价指标验证该算法在社区发现任务上的有效性，并通过与 8 个算法进行比较，验证了本算法在社区发现任务上的优越性；最后对算法进行参数分析。

4.1 数据集与基准算法

选取 3 个真实属性网络(BlogCatalog，Citeseer 和 Flickr)作为实验数据集，每个网络都包括网络拓扑结构、节点属性信息和真实的社区标签。具体来说，BlogCatalog[28] 数据集包含 6 个社区，5 196 个节点(带有 8 189 维的属性)和 171 743 条无向边。Flickr[28] 数据集包含 9 个社区，7 575 个节点(带有 12 047 维的属性)和 239 738 条无向边。Citeseer[29] 网络包含 6 个社区，3 312 个节点(带有 3 703 维的属性)和 4 732 条无向边。

与 CDEMF 进行比较的 8 个对比算法分别是 DeepWalk[10]，LINE[11]，AANE[7]，NetFS[9]，GMC[12]，NEC[13]，vGRAPH[4]，GUCD[15]。

4.2 自动确定社区个数实验分析

以 BlogCatalog，Flickr 和 Citeseer 为例，对自动确定社区数目的社区划分算法做验证实验。首先根据 CDEMF 的算法 1 得到嵌入向量 ***H***，然后运行算法 2 中的步骤 1－步骤 19，确定 I_1 次循环中出现次数最多的 k_a 和 k_b，并运行 k-means 算法计算相应的模块度 Q_1 和 Q_2，最后将模块度中较大值对应的 K 值作为最终的社区个数，如表 2 所列。

表 2 不同 k 值下的模块度和最终确定的 K 值

Table 2 Modularity against different k and final number K

Datasets	Q_t/K_t	Q_1/k_a	Q_2/k_b	final K
Citeseer[29]	0.54/6	0.44/3	**0.50/6**	6
BlogCatalog[28]	0.22/6	**0.09/6**	0.08/3	6
		0.09/6	0.09/5	6
		0.09/6	0.08/7	6
		0.08/7	0.08/3	7
Flickr[28]	0.12/9	**0.11/8**	0.11/6	8
		0.11/9	0.11/6	9

表 2 中，Q_t 为真实标签下的模块度，K_t 为正确的社区个数，k_a 为第一候选 K 值，k_b 为第二候选 K 值。Q_1 和 Q_2 分别为当 $K=k_a$ 和 $K=k_b$ 时，k-means 运行 T_2 次后计算出的最大模块度，这里取维度 $d=100$。

表 2 中，对于 Citeseer 数据集，其可能出现的 k_a 值为 3，此时的模块度为 0.44，k_b 值为 6，模块度为 0.50，非常接近真实标签下的模块，此时 $Q_1<Q_2$，因此选择 $k_b=6$ 作为最终的社区数。BlogCatalog 数据集的最大曲率和次大曲率对应的值出现的情况较多，k_a 可能为 6，7，而当 k_a 为 6 时，Q_1 为 0.09，k_b

可能为 3，5，7，Q_2 分别为 0.08，0.09，0.08。此时 $Q_1 \geqslant Q_2$，因此选择 $k_a=6$ 作为最终的 K 值。而当 $k_a=7$ 时，Q_1 为 0.08，此时 $k_b=3$，Q_2 为 0.08，所以选择 $k_a=7$ 作为最终的社区数。从 final K 可以看到，Citeseer 确定最终社区数为 6，BolgCatalog 确定最终社区数为 6 或者 7，而 Flickr 数据集确定最终社区数为 8 或者 9。

将该算法独立运行 50 次，确定社区个数的平均正确率，如图 3 所示。本文方法在 Flickr 数据集中的正确率超过了 85%，在 BlogCatalog 中的正确率超过了 90%，而在 Citeseer 中达到了 100%的正确率。由此可以看出，该方法可以自动确定社区个数，减少人工干扰的因素。

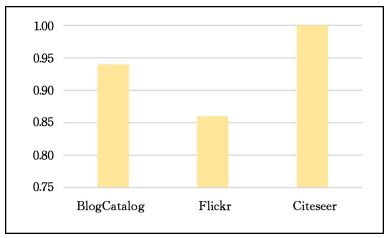


图 3 确定的社区个数的正确率

Fig. 3 Correct rate of the number of community determined

4.3 社区发现性能分析

为了进一步验证 CDEMF 算法的效果，采用 ACC，F1-Score 和 NMI 3 种评价标准来量化分析检测得到的社区结果。其中，ACC[30]用来度量社区发现结果中社区划分的正确率；F1-Score 常被用于量化真实社区标签和由算法得到的社区之间的一致程度[31]；归一化互信息 NMI 常用在社区发现或者聚类中，用于衡量预测结果和真实结果的差异。*NMI* 的值在[0，1]之间，该值越接近 1 表示预测结果越好[32]。ACC，F1-Score 和 NMI 的计算公式分别如式(20)、式(21)和式(24)所示。

$$ACC=\frac{\sum_{i=1}^{n}\ell(y_i,map(c_i))}{n} \tag{20}$$

其中，y_i 表示节点 i 的真实标签；c_i 表示算法预测得到的标签；map 函数会为预测标签 c_i 在真实标签 y_i 中找到一个最佳匹配；$\ell(a,b)$ 表示指示函数，若 $a=b$，则 $\ell(a,b)=1$，否则为 0。

$$F1\text{-}Score=\frac{2*precision*recall}{precision+recall} \tag{21}$$

其中，$precision$ 是查准率，$recall$ 是召回率，其定义如下：

$$precision=\frac{TP}{TP+FP} \tag{22}$$

$$recall=\frac{TP}{TP+FN} \tag{23}$$

其中，TP 是实际上为正，预测标签为正的个数；FP 是实际上为负，而预测标签为正的个数；FN 是实际上为正，而预测标签为负的个数[31]。

$$NMI(c,y)=\frac{2*I(c,y)}{H(c)+H(y)} \tag{24}$$

其中，$I(c,y)$ 表示真实标签 y 和预测标签 c 之间的互信息度量，$H(\cdot)$ 表示熵。

实验过程中，先用算法 1 得到每个节点的嵌入向量，再用算法 2 进行自动聚类。将所有算法重复运行 20 次，分别对评价标准取平均值。首先在 3 个数据集上比较 CDEMF 与 DeepWalk[10]，LINE[11]，AANE[7]，NetFS[9] 和 GMC[12] 的性能，结果如表 3 所列。然后在 Citeseer 数据集上将 CDEMF 算法与 GUCD[15]，NEC[13]，vGRAPH[4] 和 AANE[7] 进行实验对比，结果如图 4 所示。

表 3 各类算法在 3 个真实网络 NMI，ACC，F1-Score 上的比较

Table 3 Comparison of NMI，ACC，F1-Score of various algorithms on three real networks

	Networks	DeepWalk	LINE	NetFS	GMC	AANE	CDEMF
NMI	BlogCatalog	0.2126	0.2122	0.3264	0	0.3187	**0.3519**
	Citeseer	0.1746	0.0370	**0.3264**	0.0277	0.2085	0.3194
	Flickr	0.1711	0.1567	0.2935	0.0016	0.5624	**0.6602**
ACC	BlogCatalog	0.3925	0.4037	0.5089	0.1773	0.4715	**0.5128**
	Citeseer	0.4306	0.2342	0.5164	0.2261	0.4325	**0.5806**
	Flickr	0.3182	0.3022	0.4817	0.1182	0.6597	**0.7916**
F1-Score	BlogCatalog	0.4079	0.4239	0.5273	0.3011	0.4999	**0.5539**
	Citeseer	0.4563	0.2565	0.5102	0.3004	0.5022	**0.6102**
	Flickr	0.3492	0.3178	0.4860	0.2003	0.7160	**0.8075**

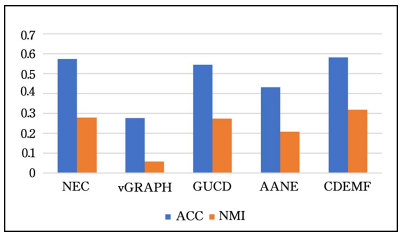


图 4 在 Citeseer 上各类算法 NMI 值、ACC 值的比较

Fig. 4 Comparison of NMI，ACC of various algorithms on Citeseer

表 3 可知，在 Citeseer 数据集中，GMC 算法的 *NMI* 值最低，只有 0.0277；NetFS 算法的 *NMI* 值最高，为 0.3264；CDEMF 仅低于 NetFS 算法，其 *NMI* 值为 0.3194。另外在 Flickr 数据集中，CDEMF 的 *NMI* 值达到了最高为 0.6602，其次是 AANE 算法，*NMI* 值为 0.5624，CDEMF 的 *NMI* 值高出 NetFS 将近 0.4。这是因为 GMC 主要针对多视图网络；AANE 融合了属性和拓扑信息，但在拓扑结构方面只利用了邻接矩阵，没有结合更多的信息；NetFS 仅仅使用拓扑信息选择出属性信息就进行聚类；而 CDEMF 深度挖掘了网络的拓扑信息，并将其与属性信息相融合，所以其在 3 个数据集上的 NMI 值优于或接近于其他算法。

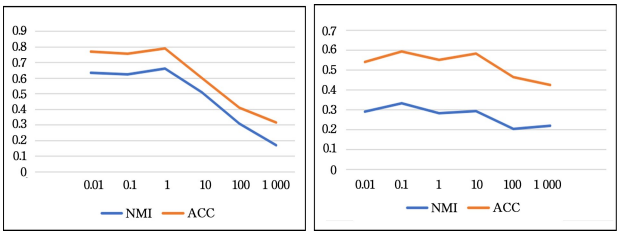
对于表 3 中的 ACC 和 F1-Score 指标，显然 CDEMF 算法都超越了其他算法。例如，CDEMF 算法的 *ACC* 值在 BlogCatalog 数据集中达到了最高，为 0.5128，其次为 NetFs 算

法，*ACC* 值为 0.508 9，而 LINE 的 *ACC* 值只有 0.403 7；在 Flickr 数据集中，CDEMF 的 *ACC* 值为 0.791 6，远远超过其他算法。在 F1-Score 指标上，DeepWalk 算法在 BlogCatalog 中取得的 F1-Score 值为 0.407 9，LINE 算法取得的 F1-Score 值为 0.423 9，CDEMF 算法取得的 F1-Score 值最高，为 0.553 9。这是因为 DeepWalk 和 LINE 只单一利用了拓扑信息，没有结合节点属性。

图 4 给出了各类算法在 Citeseer 数据集上的 NMI 值和 ACC 值的对比。从图 4 可以看到，CDEMF 算法的 *NMI* 值是最高的，超过了 0.3，其次是 NEC 算法，其 *NMI* 值不足 0.3，而 vGRAPH 算法的 *NMI* 值不足 0.1。在 ACC 值中，CDEMF 略高于 NEC 和 GUCD。可以看出，GUCD，NEC，vGRAPH 虽然融合了属性和拓扑信息，但是只利用了邻接矩阵，忽略了拓扑中丰富的信息，因此社区发现的性能不如 CDEMF。

4.4 参数选择分析

对于 BlogCatalog，λ 设定为 0.91，而对于 Citeseer 和 Flickr，分别设置 $\lambda=0.1$ 和 $\lambda=1$。为评估 λ 对性能的影响，以 Flickr 和 Citeseer 数据集为例，设定 $\lambda=[0.01, 0.1, 1, 10, 100, 1\,000]$，分别独立运行 CDEMF 算法 20 次，$\lambda$ 取不同值时算法 ACC 和 NMI 的平均值如图 5 所示。从图 5(a)中可以看到，对于 Flickr 数据集，λ 从 0.01 增加到 1，因为算法更多地考虑拓扑信息，所以提升了社团划分性能，当 $\lambda=1$ 时，CDEMF 的划分效果最好；图 5(b)中，Citeseer 数据集也有类似趋势，当 λ 接近 0.1 时，可以看到 NMI 值和 ACC 值都是接近最高的。由图 5 可知，对于 Flickr 和 Citeseer 来说，设定的 λ 一旦太大，算法会更大程度地考虑拓扑信息，而把属性的权重减少，甚至忽略属性，从而使划分效果变得非常差。另外，作为拓扑信息与属性信息的平衡参数，λ 对于 BlogCatalog 数据集的作用也是一样的。



(a)Flickr (b)Citeseer

图 5 参数设置

Fig. 5 Parameter settings

结束语 本文首先提出了一种基于矩阵分解的属性网络嵌入方法，通过矩阵分解深度融合了网络拓扑和节点属性信息，构建了包含更全面语义信息的联合低维向量。接着提出了一种基于曲率和模块度的自动聚类算法，数据仿真实验验证了其良好的社区划分效果。未来，我们准备研究动态属性网络嵌入及其与多元信息的融合，开发相应的在线和离线社区发现算法。属性网络嵌入和社区划分研究可以用于电商网络流量聚合、社交平台兴趣发现和金融领域反诈骗等方面，具有重大的理论意义和实践价值。

参 考 文 献

[1] BLONDEL V D, GUILLAUME J L, LAMBIOTTE R, et al. Fast unfolding of communities in large networks[J]. Journal of Statistical Mechanics: Theory and Experiment, 2008(10): P10008.

[2] RAGHAVAN U N, ALBERT R, KUMARA S. Near linear time algorithm to detect community structures in large-scale networks[J]. Physical Review E, 2007, 76(3): 036106.

[3] ROZEMBERCZKI B, DAVIES R, SARKAR R, et al. GEMSEC: Graph embedding with self clustering[C]// Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. 2019: 65-72.

[4] SUN F Y, QU M, HOFFMANN J, et al. vGRAPH: A generative model for joint community detection and node representation learning[C]// Advances in Neural Information Processing Systems. 2019: 514-524.

[5] YANG J, MCAULEY J, LESKOVEC J. Community detection in networks with node attributes[C]// 2013 IEEE 13th International Conference on Data Mining. IEEE, 2013: 1151-1156.

[6] JIN D, LIU Z Y, HE R F, et al. A Robust and Strong Explanation Community Detection Method for Attributed Networks[J]. Chinese Journal of Computers, 2018, 41(7): 1476-1489.

[7] HUANG X, LI J, HU X. Accelerated attributed network embedding[C]// Proceedings of the 2017 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2017: 633-641.

[8] HUANG X, LI J, ZOU N, et al. A general embedding framework for heterogeneous information learning in large-scale networks [J]. ACM Transactions on Knowledge Discovery from Data (TKDD), 2018, 12(6): 1-24.

[9] LI J, HU X, WU L, et al. Robust unsupervised feature selection on networked data[C]// Proceedings of the 2016 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2016: 387-395.

[10] PEROZZI B, AL-RFOU R, SKIENA S. Deepwalk: Online learning of social representations[C]// Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014: 701-710.

[11] TANG J, QU M, WANG M, et al. Line: Large-scale information network embedding[C]// Proceedings of the 24th International Conference on World Wide Web. 2015: 1067-1077.

[12] WANG H, YANG Y, LIU B. GMC: Graph-based multi-view clustering[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 32(6): 1116-1129.

[13] SUN H, HE F, HUANG J, et al. Network embedding for community detection in attributed networks[J]. ACM Transactions on Knowledge Discovery from Data(TKDD), 2020, 14(3): 1-25.

[14] YE Z L, ZHAO H X, ZHANG K, et al. Network representation learning algorithm based on multi-view integration[J]. Compu-

ter Science,2019,46(1):124-132.

[15] ZHANG B,YU Z,ZHANG W. Community-Centric Graph Convolutional Network for Unsupervised Community Detection [C]// IJCAI. 2020.

[16] ESTER M,KRIEGEL H P,SANDER J,et al. A density-based algorithm for discovering clusters in large spatial databases with noise[C]// KDD-96 Proceedings. AAAI,1996:226-231.

[17] RODRIGUEZ A,LAIO A. Clustering by fast search and find of density peaks[J]. Science,2014,344(6191):1492-1496.

[18] CHEN J F,ZHANG M,HE Q. NJW spectral clustering algorithm based on heuristics to determine the number of classes [J]. Computer Science,2018,45(2):474-479.

[19] TIBSHIRANI R,WALTHER G,HASTIE T. Estimating the number of clusters in a data set via the gap statistic[J]. Journal of the Royal Statistical Society:Series B(Statistical Methodology),2001,63(2):411-423.

[20] SUGAR C A,JAMES G M. Finding the number of clusters in a dataset:An information-theoretic approach[J]. Journal of the American Statistical Association,2003,98(463):750-763.

[21] ZHANG Y,MAЙDZIUK J,QUEK C H,et al. Curvature-based method for determining the number of clusters[J]. Information Sciences,2017,415-416:414-428.

[22] YANG Y,LIU H,GUAN Z,et al. CoHomo:A cluster-attribute correlation aware graph clustering framework[J]. Neurocomputing,2020,412:327-338.

[23] KUANG D,DING C,PARK H. Symmetric nonnegative matrix factorization for graph clustering[C]// Proceedings of the 2012 SIAM international conference on data mining. Society for Industrial and Applied Mathematics,2012:106-117.

[24] CALINSKI T,HARABASZ J. A dendrite method for cluster analysis[J]. Communications in Statistics-theory and Methods,1974,3(1):1-27.

[25] KRZANOWSKI W J,LAI Y T. A criterion for determining the number of groups in a data set using sum-of-squares clustering [J]. Biometrics,1988,44(1):23-34.

[26] NEWMAN M E J. Fast algorithm for detecting community structure in networks[J]. Physical Review E,2004,69(6):066133.

[27] NEWMAN M E J. Modularity and community structure in networks[J]. Proceedings of the National Academy of Sciences,2006,103(23):8577-8582.

[28] HUANG X,LI J,HU X. Label informed attributed network embedding[C]// Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. 2017:731-739.

[29] HUANG Z,ZHONG X,WANG Q,et al. Detecting community in attributed networks by dynamically exploring node attributes and topological structure[J]. Knowledge-Based Systems,2020,196:105760.

[30] PAN Y,HU G,QIU J,et al. FLGAI:a unified network embedding framework integrating multi-scale network structures and node attribute information [J]. Applied Intelligence,2020,50(11):3976-3989.

[31] GAO Y,GONG M,XIE Y,et al. Community-oriented attributed network embedding[J]. Knowledge-Based Systems,2020,193:105418.

[32] YOU X,MA Y,LIU Z. A three-stage algorithm on community detection in social networks[J]. Knowledge-Based Systems,2020,187:104822.

XU Xin-li, born in 1977, Ph.D, associate professor, is a member of China Computer Federation. Her main research interests include intelligent computing, network embedding and medical image computing.

MAO Jian-fei, born in 1976, Ph.D, associate professor, is a member of China Computer Federation. His main research interests include network visual media and mobile media technology, computer, vision and embedded system.