面向图表示社区检测的新型聚类覆盖算法

陈 洁[1,2],李 锐[1,2],赵 姝[1,2],张燕平[1,2]
(1. 计算智能与信号处理教育部重点实验室,安徽合肥 230601; 2. 安徽大学计算机科学与技术学院,安徽合肥 230601)

摘 要: 图表示社区检测使用图表示方法学习网络节点的向量表示,然后对节点向量进行聚类获得社团结构.然而经典的聚类算法在聚类节点向量时,得到的结果往往不能够体现社区的特性.提出一种新型的聚类覆盖算法,将聚类所得覆盖视为社区划分结果.首先在节点向量空间中计算得到每个簇的覆盖中心;然后根据覆盖中心到同类样本的平均距离作为覆盖半径,在向量空间中形成覆盖;最后对未覆盖的点做二次划分得到社区结构.在多个有真实和无真实标签网络的实验表明,所提出的算法可以得到更合理的社区结果.

关键词: 社区发现; 图表示; 聚类; 覆盖算法
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2020)09-1680-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2020.09.003

A New Clustering Cover Algorithm Based on Graph Representation for Community Detection

CHEN Jie[1,2], LI Rui[1,2], ZHAO Shu[1,2], ZHANG Yan-ping[1,2]
(1. *Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, Anhui University, Hefei, Anhui* 230601, *China*;
2. *School of Computer Science and Technology, Anhui University, Hefei, Anhui* 230601, *China*)

Abstract: Community detection based on graph representation learn nodes' vector representation, and then communities are obtained by clustering algorithm. However, when classical clustering algorithms often fail to reflect the characteristics of communities. Cluster cover algorithm (CCL) is proposed. CCL clusters nodes' vector into covers. A cover is viewed as a community. Firstly, the cover center of each cluster is calculated in the node vector space. Then, according to the average distance among the cover center and the same class samples as the cover radius, a cover is formed in the vector space. Finally, the nodes outside the covers are grouped into suitable cover to obtain community structure. Experiments with real and non-real tag networks show that the algorithm can get more reasonable community results.

Key words: community detection; graph represents; clustering; cover algorithm

1 引言

社区是指网络内部的节点联系相对紧密,而社区与社区之间的联系相对疏松[1].现实网络中存在大量的社区结构,正确的找到网络中的社区结构,对于揭示网络背后的信息有着重要意义.

社区检测的一个新的研究方向是通过图表示技术把网络中的关联关系数据映射成特征空间中密集连续的节点向量,并把节点向量作为 K-means[2] 等常规聚类算法的输入,从而获得社区划分的结果.如图 1 的(a)部所示,给定网络的拓扑结构,从中学习节点的特征向量,再聚类成社区结构.图表示方法由于计算复杂度低、并行化能力强、适用于传统的机器学习算法,近年来受到了广泛的关注.然而,尽管图表示算法在链接预测和节点分类等任务上表现良好,但在社区检测任务上的精度还有待提高.

交叉覆盖算法[3]作为一种经典的机器学习方法,通过在解空间中形成球形覆盖,有效的对数据进行划分.但交叉覆盖算法属于半监督学习,而社区检测属于无监督学习,无法直接对节点向量应用覆盖算法.因此,本文基于交叉覆盖的思想,提出一种新型的聚类覆盖算法,重新定义覆盖中心和覆盖半径,使其适用于图表示的社区检测.在图表示的向量空间中,计算每个簇的元首节点作为初始质心,调整质心的位置得到覆盖中心.按照距离覆盖中心的远近给每个节点一个初始标签,然后把同类节点到覆盖中心的平均距离作为半径形成球形覆盖.形成覆盖后,再结合原网络节点的结构

收稿日期:2019-09-25;修回日期:2020-04-06;责任编辑:梅志强
基金项目:国家自然科学基金项目 (No. 61602003, No. 61876001, No. 61673020),国家社科基金重大项目(No. 18ZDA032)

信息对未覆盖的点进行二次划分得到更精确的社团结构. 图 1 的(*b*)部展示了本算法的处理过程.

2　相关工作

2.1　传统社区发现算法

网络结构可以看作是一种图结构,传统的社区检测方法主要是依据图理论,针对图的邻接矩阵来提取社区特征. Fortunato 等人基于图论提出一种随机块模型[4],将图划分到一个网络中. 张等人利用图的局部链接相似性进行聚类得到社团结构[5]. 基于图论的社区检测方法直观,具有坚实的理论基础.

从网络局部结构开始优化和扩展到全局网络也是一种可靠的手段. LFM 算法[6]随机选取一个节点作为初始种子,从该种子节点出发,向外扩展,逐步构建一个社区. Louvain 算法[7]开始时把每节点作为独立的社团,然后从邻居节点中选取模块度增益最大的节点加入,逐步合并直到遍历网络中所有节点. 於志勇等人则根据种子影响力来对种子节点扩展发现社区[8].

网络的边也可看作一种路径,网络的个体可以通过路径传播信息. Raghavan 等人基于这种思想提出了一种标签传播算法[9]. 传统方法一般直接作用于网络,从网络的邻域信息或局部结构来得到社团的解.

2.2　图表示方法

图表示算法通过把网络中每个节点或边表示成低维空间的向量,同时提取网络节点的特征,即相似性大的节点在向量空间中距离更近,相似性小的节点距离更远[10].

Deep walk 算法[11]作为经典的网络嵌入的方法,通过在网络中均匀的随机游走,获得大量随机游走序列,送入神经网络模型进行概率建模,输出每个节点的向量表示. Node2vec[12]则改进了 Deep walk 算法的随机游走方式,通过 p,q 两个参数控住游走的概率. 随机游走的方式在处理网络扩散任务上表现出了良好的性能[13]. SDNE[14]则是一种基于深层神经网络的表示学习模型,通过编码器和解码器对网络相似度关系进行建模,输出深层自编码器的中间层作为向量表示.

2.3　交叉覆盖算法

覆盖算法可看作是一个多层前向网络分类器. 假设数据集 $D=\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$. 覆盖算法先将样本映射到球形领域,以样本 x_i 为圆心,Δ_i 为半径构造一个球形覆盖,把同属于 x_i 类样本包裹在覆盖内,覆盖中不包含其他类别数据. 根据 M-P 神经元的球形几何意义[15],一个球形覆盖即可看成前向神经网络的一个神经元. 覆盖算法的优点是网络简单,可解释性强.

交叉覆盖算法是覆盖算法最初的模型,所谓的交叉覆盖即交替覆盖. 将样本投射到球形领域后,随机选择一个样本点开始覆盖,将其类别记为 k^1,以距离覆盖中心最近的异类样本为界,以界内最远的同类点到覆盖中心的距离为半径求出一个领域 C^1,只覆盖 k^1 中的点. 然后将被 C^1 覆盖的点删去,对余下的点求另一个领域 C^2,只覆盖 k^2 的点,然后将被 C^2 覆盖的点删去,…. 如此交叉进行覆盖,最终获得一个覆盖集合 $C=\{C_1^1,C_2^1,\cdots,C_{s1}^1,C_1^2,C_2^2,\cdots,C_{s2}^2,\cdots,C_1^k,C_2^k,\cdots,C_{sk}^k\}$.

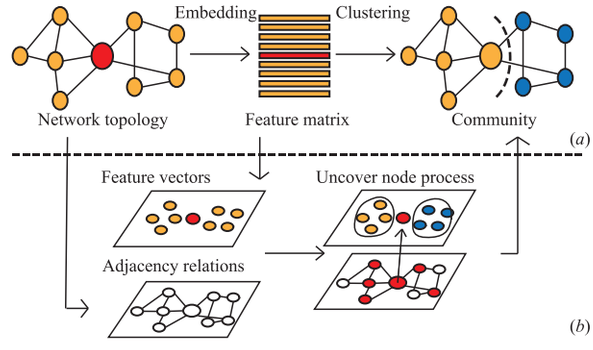

图1　图表示社区检测算法流程

3　算法描述

本文使用经典的图表示算法 Deep walk 作为对网络进行预处理的算法,以学习网络节点的向量表示. 将向量作为节点特征属性输入到聚类覆盖算法得到社区结果.

3.1　相关定义

3.1.1　问题定义

通常一个无向无权的网络被看作是一个图 $G=(V,E)$,$V=\{v_1,v_2,\cdots,v_n\}$ 是网络中点的集合,E 是边的集合. 基于图表示的社区检测,用图表示的方法学习网络节点的向量集合 P,作为节点的特征输入到聚类算法中得到社区. 一个社区通常被定义为一个节点的集合 $C=\{v_1,v_2,\cdots,v_m\}$,非重叠社区发现旨在识别满足 $C=\{C_1,C_2,\cdots,C_k\}$ 和 $C_1\cup C_2\cup\cdots\cup C_k\in V$ 两个条件的社团.

3.1.2　符号定义

定义 1(点密度 ρ)　在网络的向量空间中,节点 v_i 的密度表示为节点 v_i 附近点的稠密程度. 计算公式如下:

$$\rho_i=\sum_{v_j\in V}\chi(\mathrm{dist}(\boldsymbol{p}_i,\boldsymbol{p}_j)-d_c) \tag{1}$$

$$\chi(x)=\begin{cases}1, & \text{if}\quad x\leqslant 0\\ \exp\left(-\left(\dfrac{\mathrm{dist}(\boldsymbol{p}_i,\boldsymbol{p}_j)}{d_c}\right)^2\right), & \text{if}\quad x>0\end{cases} \tag{2}$$

d_c 表示截断距离,设数据点的平均邻居点数占数据集总样本数 N 的比例为 $P\in(0,1)$,将 $M=N(N-1)$ 个点与点之间的距离按从小到大排序记为 $d_1\leqslant d_2\leqslant\cdots\leqslant d_M$. 则 $d_c=d_{f(PM)}$,$f(PM)$ 为 PM 四舍五入后取整数. ρ_i 表示点 v_i 的密度,$\boldsymbol{p}_i$ 和 $\boldsymbol{p}_j$ 为点 v_i 和 v_j 的向量表示. 距离 v_i 小于 d_c 的点对 v_i 密度贡献为 1,大于 d_c 的点用一个高斯函数来表示,距离越远其值越小.

定义 2(分隔距离 δ) 在网络的向量空间中,点 v_i 的分隔距离 δ_i 表示点 v_i 与任何比点 v_i 密度更大点的最小距离,计算方式如式(3)所示. 对于密度最大的点来说,$\delta_i = \max\ \mathrm{dist}(\boldsymbol{p}_i, \boldsymbol{p}_j)$.

$$\delta_i = \min_{j:\rho_j > \rho_i} \mathrm{dist}(\boldsymbol{p}_i, \boldsymbol{p}_j) \tag{3}$$

定义 3(元首结点 $\boldsymbol{\alpha}$) 在网络的表示向量空间中,可以唯一代表一个社区的节点. 元首节点具有两个性质:(1)元首节点本身的密度很大,被密度不高于它的邻居节点包围着;(2)元首节点与其他密度高于它的节点距离相对更大. 元首节点的性质来源于 laio 等人在 2014 年提出的 DPC 算法(密度峰聚类)[16].

3.2 聚类覆盖算法

3.2.1 覆盖中心和覆盖半径

由定义 3 中元首节点的性质可以得出,元首节点是密度和分隔距离都相对较大的节点,本文通过计算密度 ρ 和分隔距离 δ 的乘积 γ 来判断元首节点. 显然,对于 ρ 值和 δ 值都很小的节点不可能成为元首节点. 假设 ρ 值和 δ 值都低于 80% 的节点不具有成为元首节点的潜质,所以本文的方法只对 ρ 值和 δ 值都超过 80% 的点计算 γ 值. 然后将 γ 按元素从小到大排序构成一个排序图 γ^s. I 表示排序后对应下标的集合,如给定三个数 $\gamma_2 > \gamma_1 > \gamma_3$,排序后 $\gamma_1^s = \gamma_2, \gamma_2^s = \gamma_1, \gamma_3^s = \gamma_3, I = \{2,1,3\}$. γ 计算方法如下:

$$\gamma_i = \rho_i * \delta_i \tag{4}$$

图 2 为 karate 网络节点经过排序后的 γ^s 图,由图可以看出元首节点 γ 值较大,一般位于右上角,图中红色的点所示. 而普通样本点由于 γ 较小,几乎分布在一条直线上,图中蓝色的点所示. 首先,拟合一条直线穿过 γ^s 图首尾两点 $(1,\gamma_1^s)$和$(|I|,\gamma_{|I|}^s)$,如图 2 所示. 直线公式如下:

$$y = \frac{\gamma_{|I|}^s - \gamma_1^s}{|I| - 1} x + \frac{|I| \cdot \gamma_1^s - \gamma_{|I|}^s}{|I| - 1} \tag{5}$$

然后求排序图中到该直线距离最远的点坐标(i,γ_i^s).

$$(i,\gamma_i^s) = \underset{x,y}{\mathrm{argmax}} \left(\frac{|Ax + By + C|}{\sqrt{A^2 + B^2}} \right) \tag{6}$$

其中 A,B,C 是直线的系数. 所以元首节点的定义如下:

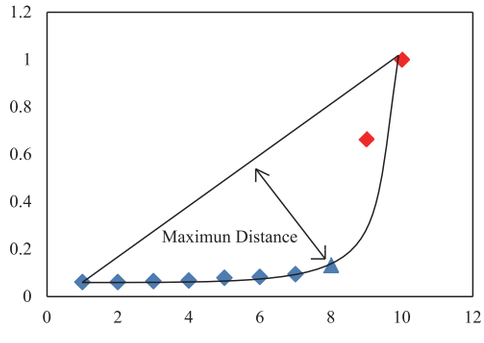

图2 karate网络的元首节点选择图

$$\alpha = \{v_x \mid x \in [I_{i+1}, I_{|I|}]\} \tag{7}$$

得到元首节点后并不能直接作为覆盖中心. 网络中的节点通过图表示后在向量空间中会以簇状结构分布,而元首节点作为密度最大的点不一定位于簇的中心. 为了使球形覆盖可以尽可能的覆盖较多的节点,需要调整覆盖中心的位置. 本文采用 K-means 算法的思想,把元首节点作为初始质心,每个样本点根据欧式距离的大小赋予最近的质心相同的标签,再根据同类样本的向量均值调整质心的位置,最后再依据到质心的距离重新赋予样本标签,不断迭代,直到质心收敛,把收敛后的质心作为覆盖中心. 具体步骤如下:

步骤 1 对每一个样本 v_i,赋予最近质心相同的标签:

$$L_i := \underset{j}{\mathrm{argmin}} \| \boldsymbol{p}_i - \boldsymbol{\mu}_j \|^2 \tag{8}$$

步骤 2 L_i 表示节点 v_i 的标签,$\boldsymbol{\mu}_j$ 为质心,对每一个类 j,根据同类样本的均值重新计算类质心 $\boldsymbol{\mu}_j$:

$$\boldsymbol{\mu}_j := \frac{\sum_{i=1}^{m} 1\{L_i = j\} \boldsymbol{p}_i}{\sum_{i=1}^{m} 1\{L_i = j\}} \tag{9}$$

重复步骤 1 和步骤 2,直到质心的位置不再改变. 把收敛后的质心 $\boldsymbol{\mu}_j$ 作为覆盖中心 $\boldsymbol{s}_j$,得到覆盖中心集合 $S = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_k\}$. 质心收敛后,会赋予每个样本一个初始标签,本文以同类标签到覆盖中心的平均距离作为覆盖半径. 计算方式如式(10)所示,L_x 为样本 v_x 的标签,$\boldsymbol{s}_j$ 为覆盖中心.

$$r_j = \frac{\sum_{L_x = j} \mathrm{dist}(\boldsymbol{p}_x, \boldsymbol{s}_j)}{\sum_{L_x = j} 1} \tag{10}$$

3.2.2 未覆盖节点处理

3.2.1 节在向量空间中形成覆盖后,认为覆盖内的样本属于该社团核心节点,依据距离信息可以做出有效划分. 覆盖外的样本属于该社团的非核心节点,由于远离覆盖中心,也受到网络表示学习算法性能的影响,只依据空间距离信息无法做出合理的判断,所以对未覆盖样本除了考虑距离信息外,增加另外一种信息,样本的邻域信息. 节点的邻域大多属于同一个社团,则该样本极有可能属于该社团. 把样本邻域信息作为距离信息的权重辅助距离信息进行二次决策,可以对未覆盖节点做出有效的划分.

定义 4(未覆盖点邻接矩阵 $\boldsymbol{B}$) 假设网络中有 n 个节点,给定网络的邻接矩阵 $\boldsymbol{A} \in \mathbf{R}^{n \times n}$,划分覆盖后的未覆盖点集合 U. 根据式(11)构建未覆盖点邻接矩阵 $\boldsymbol{B} \in \mathbf{R}^{m \times n}$. 矩阵 $\boldsymbol{B}$ 只保留邻接矩阵 $\boldsymbol{A}$ 中未覆盖点所在的行.

$$\boldsymbol{B}_{ij} = \boldsymbol{A}_{ij}, \quad v_i \in U \tag{11}$$

定义 5(标签矩阵 $\boldsymbol{L}$) 覆盖集合 $C' = \{c_1', c_2', \cdots,$

$c'_k\}$ 表示划分的 k 个覆盖. 标签矩阵 $\boldsymbol{L}\in\mathbf{R}^{n\times k}$ 表示 n 个节点对应 k 个覆盖的标签,若点 v_i 已划分在覆盖 c_j'中,则 $\boldsymbol{L}_{ij}=1$,否则为 0,对于未覆盖点来说都是 0.

定义 6(距离矩阵 $\boldsymbol{D}$)　距离矩阵 $\boldsymbol{D}\in\mathbf{R}^{m\times k}$ 的行由未覆盖点组成,列由覆盖中心组成,每个元素表示该点与各个覆盖中心的相似程度,由欧式距离计算得到. 由于距离越近两个点越相似,于是这里取距离的倒数表示点的相似程度. 距离矩阵 $\boldsymbol{D}$ 构造如式(12)所示,其中 $\boldsymbol{p}_i$ 是未覆盖点的向量表示:

$$\boldsymbol{D}(i,j)=\frac{1}{\operatorname{dist}(\boldsymbol{p}_i,\boldsymbol{s}_j)} \tag{12}$$

权重矩阵 $\boldsymbol{W}\in\mathbf{R}^{m\times k}$ 由矩阵 $\boldsymbol{B}$ 和矩阵 $\boldsymbol{L}$ 通过矩阵相乘得到. 权重矩阵 $\boldsymbol{W}$ 中元素值表示各个未覆盖点的邻域节点在各个覆盖中的比重. 得到距离矩阵 $\boldsymbol{D}$ 和权重矩阵 $\boldsymbol{W}$ 后,将矩阵 $\boldsymbol{D}$ 和矩阵 $\boldsymbol{W}$ 的哈达玛积做为决策矩阵 $\boldsymbol{H}$. 决策矩阵 $\boldsymbol{H}$ 会得到未覆盖样本点到各个覆盖的一个得分. 把得分最高的点划分到相应的覆盖中,即更新标签矩阵 $\boldsymbol{L}$ 中该点对应覆盖的值为 1,然后再更新矩阵 $\boldsymbol{B}$ 该点所在行对应的值都为 0,以保持矩阵 $\boldsymbol{B}$ 的行列数不变. 把更新完的矩阵 $\boldsymbol{L}$ 和矩阵 $\boldsymbol{B}$ 再次相乘得到新的权重矩阵 $\boldsymbol{W}'$,进入下一轮迭代,每次只更新一个节点的标签,直到标签矩阵 $\boldsymbol{L}$ 中所有未覆盖点全部更新完. 输出标签矩阵 $\boldsymbol{L}$ 即为所求的社团结构. 即使有未覆盖点的邻域节点都不在覆盖中,随着邻域节点的先行划分最终也会划分到覆盖中. 算法 1 为聚类覆盖算法的总体描述。

算法 1　聚类覆盖算法(CCL)

输入　网络节点集 V 和节点向量集 p
输出　标签矩阵 $\boldsymbol{L}$
for $i=1$ to $|V|$do:
　根据式(1)计算局部密度 ρ_i;
　根据式(3)计算分隔距离 δ_i;
end for
$\rho'=\dfrac{\rho_i-\min(\rho)}{\max(\rho)-\min(\rho)}$
$\delta'=\dfrac{\delta_i-\min(\delta)}{\max(\delta)-\min(\delta)}$
过滤密度 ρ'和分隔距离 δ'相对较小的点;
对保留的点根据式(4)计算 γ 值;
$(\gamma^s,I)=\text{sortAscending}(\gamma)$;
在$(1,\gamma_1^s)$和$(|I|,\gamma_{|I|}^s)$之间连接一条直线;
$i\leftarrow$找到距离直线最远的一点(i,γ_i^s);
$\alpha\leftarrow\{v_x\,|\,x\in[I_{i+1},I_{|I|}]\}$;
$\boldsymbol{\mu}\leftarrow\alpha$//元首节点作为初始质心;
repeat
　根据式(8)计算每个样本的标签 ;
　根据式(9)调整质心的位置;
until 质心的位置不再改变
$S\leftarrow\boldsymbol{\mu}$//收敛后的质心作为覆盖中心;
根据式(10)计算每个覆盖中心对应的覆盖半径 r_j;
for each $v_i\in V$ do
　for each $\boldsymbol{s}_j\in S$ do
　　if dist$(\boldsymbol{p}_i,\boldsymbol{s}_j)<r_j$:
　　$C_j'\leftarrow v_i$
　　else $U\leftarrow v_i$
　end for
end for
初始化矩阵 $\boldsymbol{B}^0$ 和矩阵 $\boldsymbol{L}^0$
根据式(12)构造距离矩阵 $\boldsymbol{D}$
for k:1 to $|U|$ do
　$\boldsymbol{W}^k=\boldsymbol{B}^{k-1}\cdot\boldsymbol{L}^{k-1}$
　$\boldsymbol{H}^k=\boldsymbol{W}^k*\boldsymbol{D}$
　$(x,y)=\arg\max_{(i,j)}\boldsymbol{H}^k(i,j)$
　$\boldsymbol{B}^k\leftarrow\boldsymbol{B}_x^{k-1}=0$//更新 $\boldsymbol{B}$ 中该点所在行的所有值为 0
　$\boldsymbol{L}^k\leftarrow\boldsymbol{L}_{xy}^{k-1}=1$//更新 $\boldsymbol{L}$ 该点对应的值为 1
end for
return $\boldsymbol{L}^k$

3.3　基于聚类覆盖的图表示社区检测算法

图表示社区检测用图表示学习节点的低维向量表示,并作为节点特征输入到常规聚类算法中获得社区结构. 但常规聚类算法对向量的聚类无法很好地体现社团特性. 因此,本文改进经典的交叉覆盖算法,提出一种新的聚类覆盖算法(CCL)用于图表示社区检测中对向量的聚类. 并结合 Deep walk 图表示算法提出基于聚类覆盖的图表示社区检测算法(Graph represents Community Detection based on CCL, CD-CCL). 先用 Deep walk 算法来预处理网络,学习节点的向量表示. 然后用新的聚类覆盖算法来对向量进行聚类,挖掘网络中的社区结构. 算法 2 为 CD-CCL 算法的流程.

算法 2　基于聚类覆盖的图表示社区检测算法(CD-CCL)

输入　网络点的集 V 和边的集合 E
输出　社区结构 C
$P=\text{Deepwalk}(V,E)$
$\boldsymbol{L}^k=\text{CCL}(V,P)$
$C\leftarrow\boldsymbol{L}^k$
return C

4　实验分析

本文选取了真实有标签的网络和无标签网络作为实验对象. 在有标签的网络上,每个节点都有一个真实的社团标签,本文对有标签网络用归一化互信息(NMI),召回率(recall)和精度(precision)进行评价,在无标签的网络由于没有真实的社团结构,本文只讨论社团模块度(Q).

4.1　数据集

本次实验共选择了 8 个公用网络数据集,包括 6 个

规模不同的有真实标签网络和2个无真实标签网络,网络具体信息如表1所示.6个真实有标签网络分别是空手道网络俱乐部(Zachary's karate club),海豚网络(dolphin social network),足球联盟网络(American college football),美国政治书籍网络(books about US politics).DBLP数据集和Amazon数据集属于重叠社团数据集,为了得到精确地非重叠社团划分,本文对DBLP和Amazon数据集进行采样.为了保证采样子网络的连通性,本文随机抽取一个节点,然后采样该节点的一阶邻域节点和二阶邻域节点作为子网.两个无标签网络是电子邮件通信网络(Email communication network),网络科学合作者网络(co-authorships in network science).

表1 数据集

Network	$\|V\|$	$\|E\|$	Class
Karate	34	78	2
Dolphin	62	159	2
Politics	105	441	3
Football	115	613	12
Email	1133	5454	unknow
Net-science	1589	2742	unknow
Amazon	10361	15269	70
DBLP	27783	54083	1685

4.2 对比算法

本文选择与当前比较流行的几个社团发现算法和经典的聚类算法进行比较.

LPA[9]:一种标签传播社区发现算法.

louvain[7]:经典的基于模块度优化的贪心算法.

SSC[17]:一种最新的基于混合子空间的稀疏性编码网络社区检测.

K-means[1]:一种经典的基于划分的聚类算法.

DPC[16]:一种新型的基于密度传播的密度峰聚类算法.

层次聚类:一种经典的自底向上层次树聚类算法.

4.3 评价指标

4.3.1 归一化互信息

归一化互信息(NMI)评价所划分社团与真实社团结构之间的准确度,在聚类中,也可以度量两个聚类结果的相近程度,其值域为0到1.值越高代表划分的结果越准确.NMI的定义如下:

$$\mathrm{NMI}=\frac{-2\sum_{i=1}^{k_t}\sum_{j=1}^{k_p}n_{ij}^{tp}\log\left(\frac{n_{ij}^{tp}n}{n_i^t n_j^p}\right)}{\sum_{i=1}^{k_t}n_i^t\log\left(\frac{n_i^t}{n}\right)+\sum_{j=1}^{k_p}n_j^p\log\left(\frac{n_j^p}{n}\right)} \tag{13}$$

其中n是网络中节点总数,t是真实社团结构,p是所划分的社团结构.k_t是真实社团个数,k_p是所划分的社团个数.而n_i^t和n_i^p分别表示t和p中的一个社团.n_{ij}^{tp}表示社团i与社团j的重叠点数.

4.3.2 模块度

社团模块度(Q)由一种衡量社区划分强度的方法,主要用来衡量社团结构未知的网络.模块度值由所划分的社团内部边数量与社团间边数量的比值决定,模块度值越高,则其划分的社团质量越好,但不超过1.其定义如下:

$$Q=\frac{1}{2m}\sum_{ij}\left[\boldsymbol{A}_{ij}-\frac{k_ik_j}{2m}\right]\partial(C_i,C_j) \tag{14}$$

$\boldsymbol{A}_{ij}$表示图邻接矩阵,k_i和k_j表示节点i和节点j的度,$\partial(C_i,C_j)$函数表示若节点i和节点j在一个社团内返回1,不在返回0.

4.3.3 准确率和召回率

准确率和召回率用来评价所划社区中每个社区和真实的社区相比较.准确率是指预测的一个社区中,划分正确的样本在预测社区中的比重.召回率是指预测的一个社区中,划分正确的样本在真实社区中的比重.本文取所有社区的precision和recall的均值作为最终的实验结果.如式(15)、(16)所示,其中C^{D}为预测的社团划分,C^{R}为真实的社团划分.

$$\mathrm{percision}(C_i^{\mathrm{D}})=\frac{C_i^{\mathrm{D}}\cap C_j^{\mathrm{R}}}{C_i^{\mathrm{D}}} \tag{15}$$

$$\mathrm{recall}(C_i^{\mathrm{D}})=\frac{C_i^{\mathrm{D}}\cap C_j^{\mathrm{R}}}{C_j^{\mathrm{R}}} \tag{16}$$

4.4 实验结果

4.4.1 分析覆盖半径

本文对覆盖半径计算是以同类标签到覆盖中心的平均距离来确定.该部分通过分析覆盖半径的大小对社区结果的影响来证明这样计算的合理性.本节选取三个数据集来证明把平均距离作为半径在保持算法性能的同时可以最小化计算复杂度,分别是海豚网络(dolphin),政治书籍网络(politics)和足球网络(football).

实验从给每个样本一个初始标签开始,确定不同半径对NMI值,准确率和召回率指标造成的影响.半径采用从零开始梯度增加的方法来确定半径对实验结果的影响.如图3(a)所示,横坐标从0开始以0.5的步长遍历半径的值到同类标签样本到中心点最大距离为止.纵坐标为三个评价指标对应的具体数值.

综合图3可以发现,当半径小于一定阈值时,算法的性能几乎不受影响,超过该阈值后,算法的性能开始下降.覆盖半径过大会导致一些样本的决策出现错误,影响了算法性能,所以需要合理控制半径.通过计算发现同标签样本到中心点的距离均值一般是小于阈值的大小,图中对均值点做了标注,可以发现,均值点一般位于阈值之后.所以以距离均值作为半径既可以保持算法的性能,同时也可以最小化需要在再次决策的样本数,减小了计算的复杂度.

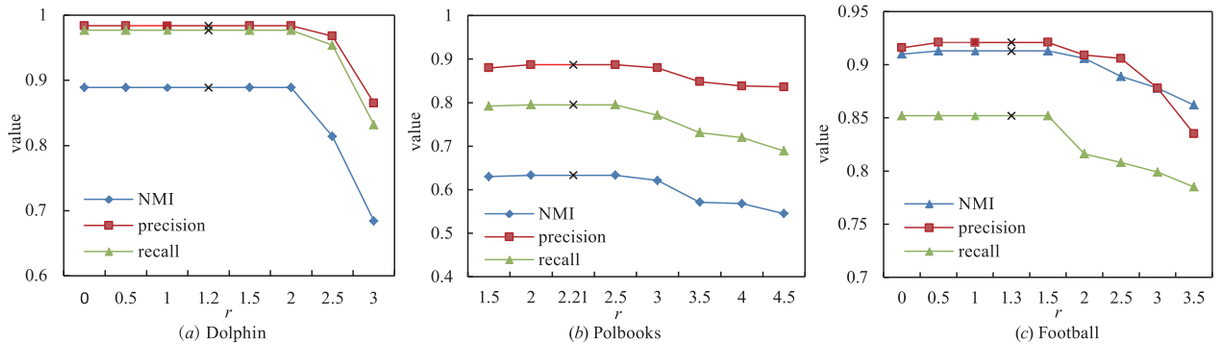

图3　实验效果随覆盖半径变化图

4.4.2　与传统聚类算法比较

该部分将本文提出的 CCL 算法与已有的比较典型的聚类算法比较,包括 K-means 算法,密度聚类 DPC 和层次聚类. 统一使用 Deep walk 图表示法对网络进行预处理转成向量,然后对向量进行聚类. Deep walk 中基本参数设置为表示向量的维度为 64,从每个节点开始的游走数目为 10,游走的步长为 40,滑动窗口的尺码为 5. 图 4 中(*a*)和(*b*)分别展示了本文的 CCL 算法与这 3 种聚类算法在 NMI 值和 precisions 值的比较. 由于 K-means 算法需要提前设定聚类数,本文按照真实社团数给定 K-means 参数值. DPC 算法的优势在于发现任意形状的聚类簇,缺点是需要人工选择聚类中心,在处理社团较多的网络时容易出错. 层次聚类构建层次树时每次合并两个最近的簇,造成复杂度过高,无法应用在 Amazon 和 DBLP 这种大网络上,为了方便观察,层次聚类在 Amazon 和 DBLP 的结果以一个极小值 0.1 给出.

图 4 的实验表明,本文的 CCL 算法在处理图表示的向量聚类上要普遍优于其他典型的聚类算法. 由图 4(*a*)和图 4(*b*)可以看出,本文算法与 K-means 算法比较时,尽管 K-means 算法按照真实的社团数设定的参数,效果比未知的情况下有所提升,在 NMI 值上和精确率上还是要稍差于本文的聚类覆盖算法. 与 DPC 算法比较可以看出我们的算法是明显好于 DPC 聚类算法. 与层次聚类比较时,层次聚类在小数据集上 NMI 比 CCL 稍差,精确率上与 CCL 算法几乎持平,但层次聚类有一个严重的缺点是其复杂度太高,造成适用性不高. 本文提出的 CCL 算法相较于传统的聚类算法,对图表示特征的聚类可以得到更合理的社区结构.

4.4.3　与主流社团发现算法比较

本部分把基于聚类覆盖的图表示社区检测算法与一些主流的社区发现算法在有标签网络和无标签网络的比较. 有标签网络有真实的社区结构,通过归一化互信息 NMI 和召回率 Recall 来评价算法划分的结果与真实结果的差异. 在无标签网络上,由于没有真实的社团结构,用模块度 Q 来评价划分的社团强度. 本文选取了 8 个大小不同的数据集,包括 6 个有标签的数据集和 2 个无标签的数据集.

图 4(*c*)和图 4(*d*)分别为不同算法在 6 个有真实标签的数据集的实验. 图 4(*c*)图为 NMI 值的对比,图 4(*d*)图为召回率 Recall 的比较. 从图中可以看出本文提出的算法的实验结果要普遍优于其他对比算法. 其中在 karate 网络上,本文与 SSC 算法都取得了完全正确的划分,在其他 3 个数据集上的,CD-CCL 算法要领先于 SSC. 总体而言,本文的算法在有标签的真实社团划分的准确性上优于其他算法.

对于无标签网络,本文使用两个无标签数据集,分别是西班牙大学电子通讯网络和互联网科学合作者网络. 结果展示在表 2 里. 从表 2 可以发现,本文的算法在 Email 网络上取得了 0.562,要明显优于其他算法. 在 Network science 网络上,本文的算法好于 LPA 算法,略低于 Louvain 算法和 SSC 算法,其中 Louvain 算法是针对模块度优化的,在模块度上有一定的优势. 在未知网络的划强度上,本文的 CD-CCL 算法也取得了不错的表现.

表 2　无标签网络实验结果

Network	LPA	louvain	SSC	CD-CCL
Email	0.532	0.546	0.544	0.562
N-science	0.937	0.957	0.958	0.955

4.4.4　可扩展性分析

本节的实验环境设置为:Window 10 操作系统,inter(R) Core(TM)i5-7300HQ 2.5GHz CPU,8G 内存. 实验基于 Python3.6 语言编写实现. 本文提出的 CD-CCL 算法分为离线和在线两个阶段,其中离线阶段用 Deepwalk 算法预处理网络,对网络进行特征提取. 在线阶段用本文提出的 CCL 算法进行聚类. 表 3 为算法在线阶段不同数据集上的运行时间,可以看出,随着网络规模的增大,计算的时间消耗大部分来源于计算各个覆盖的覆盖中心上. 由于这部分需要对网络中全部节点计算距离,通过距离的远近来确定点密度的大小,因此带来较

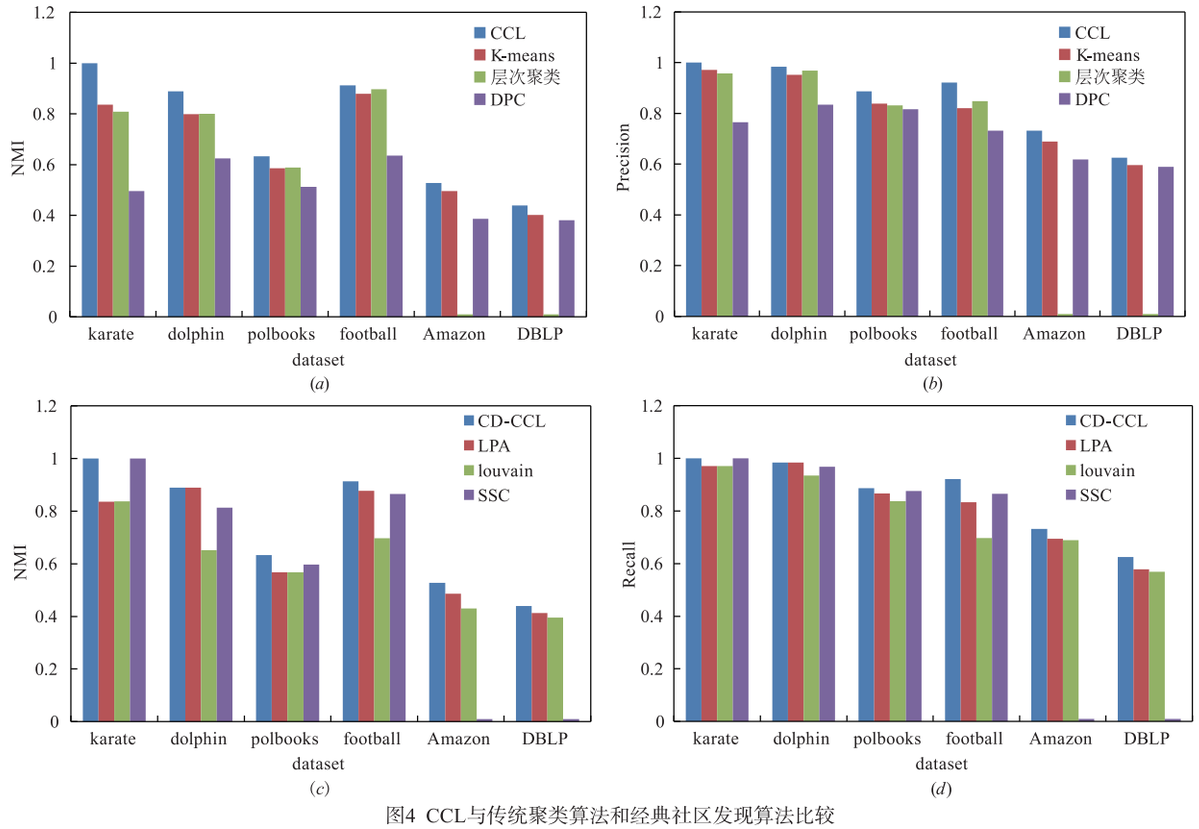

图4 CCL与传统聚类算法和经典社区发现算法比较

高的时间消耗. 但由于能被选做覆盖中心的点非常少，取决于网络中的社团数，大部分节点由于密度较低，并不具有成为覆盖中心的潜质，在计算大规模网络时可采用过滤淘汰的方式减小这部分节点的计算消耗. 实验发现，有成为覆盖中心潜质的点往往具有很高的度，即节点的邻居节点数较多. 因此本文通过计算所有节点的度的平均数作为阈值，过滤掉度小于阈值的节点，不参与密度的计算. 由表 3 可以看出过滤后的时间消耗情况，在保证了实验效果的前提下，大大减小了计算覆盖中心的时间消耗，总的时间消耗也随之降低. 随着网络规模的增大，时间消耗减少的更加明显，也使本文的算法可以更好的适应较大规模网络的计算，使算法具有良好的可扩展性.

表 3 在线阶段时间消耗

时间 / dataset	过滤前		过滤后	
	计算中心	总时间	计算中心	总时间
Karate	0. 021s	0. 111s	0. 005s	0. 107s
Dolphin	0. 069s	0. 171s	0. 032s	0. 134s
Football	0. 259s	0. 451s	0. 247s	0. 654s
Polbooks	0. 217s	0. 366s	0. 046s	0. 177s
Email	24. 33s	41. 69s	7. 648s	15. 85s
N-science	42. 16s	66. 31s	7. 210s	18. 325s
Amazon	0. 558h	0. 660h	0. 157h	0. 292h
DBLP	2. 46h	3. 28h	0. 285h	0. 561h

5 结束语

随着网络的发展，网络的规模越来越大，传统的社区发现算法采用邻接矩阵来做社区发现，往往会受到复杂度的制约. 基于模块度优化的算法虽然可以快速处理大型数据集，但会陷入局部最优，在社团精确率上得不到最优解. 网络表示学习通过把网络表示成向量的形式降低了计算复杂度，本文用网络表示的方法对网络进行预处理，在对向量用本文提出的基于聚

类覆盖算法对向量聚类得到高精确率的社团结构. 网络表示学习的方法有很多，本文仅是使用了经典的 Deep walk 算法处理无向无权网络. 对加权网络和带有语义的网络等，可以替换其他的网络表示方法，所以模型的提升空间还很大，将来会针对不同形式的网络进行研究，来得到更稳定和更多样化的模型.

参考文献

[1] Newman M E J. Detecting community structure in networks [J]. The European Physical Journal B，2004，38(2)：321 – 330.

[2] Hartigan J A，Wong M A. Algorithm AS136：A *k*-means clustering algorithm[J]. Journal of the Royal Statistical Society. Series C (Applied Statistics)，1979，28(1)：100 – 108.

[3] 张铃,张钹. 多层前向网络的交叉覆盖设计算法[J]. 软件学报,1999,10(7):737 – 742.
Zhang L, Zhang B. An alternative covering design algorithm of multi-layer neural networks[J]. Journal of Software,1999. 10(7):737 – 742. (in Chinese).
[4] Fortunato S, Hric D. Community detection in networks: A user guide[J]. Physics Reports,2016,659:1 – 44.
[5] 张桂杰,张健沛,杨静,等. 基于链接相似性聚类的重叠社区识别[J]. 电子学报,2015,43(7):1329 – 1335.
Zhang GJ, Zhang JP, Yang J, et al. Overlapping community detection based on link similarity clustering[J]. Acta Electronica Sinica,2015,43(7):1329 – 1335. (in Chinese).
[6] Li H J, Bu Z, Li A, et al. Fast and accurate mining the community structure: integrating center locating and membership optimization[J]. IEEE Transactions on Knowledge and Data Engineering,2016,28(9):2349 – 2362.
[7] DeMeo P, Ferrara E, Fiumara G, et al. Generalized Louvain method for community detection in large networks[A]. Intelligent Systems Design and Applications[C]. Cordoba, Spain: IEEE,2011. 88 – 93.
[8] 於志勇,陈基杰,郭昆,等. 基于影响力与种子扩展的重叠社区发现[J]. 电子学报,2019,47(01):155 – 162.
Yu ZY, Chen JJ, Guo K, et al. Overlapping community detection based on influence and seeds extension[J]. Acta Electronica Sinica,2019,47(01):155 – 162. (in Chinese).
[9] 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 – 036118.
[10] Cui P, Wang X, Pei J, et al. A survey on network embedding[J]. IEEE Transactions on Knowledge and Data Engineering,2018,31(5):833 – 852.
[11] Perozzi B, Al-Rfou R, Skiena S. Deepwalk: Online learning of social representations[A]. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. New York: ACM,2014. 701 – 710.
[12] Grover A, Leskovec J. node2vec: Scalable feature learning for networks[A]. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. San Francisco: ACM,2016. 855 – 864.
[13] 陶冶,张书奎,张力,等. 移动感知器网络中基于随机游走和协作关系的任务分发算法[J]. 电子学报,2019,47(8):1601 – 1611.
Tao Y, Zhang SK, Zhang L, et al. Task distribution algorithm based on random walk and cooperative relationship in mobile sensor networks[J]. Acta Electronica Sinica, 2019,47(8):1601 – 1611. (in Chinese).
[14] Wang D, Cui P, Zhu W. Structural deep network embedding[A]. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. San Francisco: ACM,2016. 1225 – 1234.
[15] 张铃,张钹. M-P 神经元模型的几何意义及其应用[J]. 软件学报,1998,9(5):334 – 338.
Zhang L, Zhang B. A geometrical representation of M-P neural model and its applications[J]. Journal of Software, 1998,9(5):334 – 338. (in Chinese).
[16] Rodriguez A, Laio A. Clustering by fast search and find of density peaks[J]. Science,2014,344(6191):1492 – 1496.
[17] Tian B, Li W. Community detection method based on mixed-norm sparse subspace clustering[J]. Neurocomputing,2018,275:2150 – 2161.

作者简介

陈　洁　女,1982 年 10 月出生,安徽巢湖人. 安徽大学副教授,硕士生导师. 主要研究方向为机器学习,粒计算和三支决策.
E-mail:chenjie200398@163.com

李　锐　男,1996 年 9 月出生,安徽定远人,安徽大学计算机科学与技术学院硕士生,主要研究方向为机器学习,社团发现.
E-mail:lirui1101998040@163.com

赵　姝　女,1979 年 10 月出生,安徽巢湖人. 安徽大学教授,博士生导师. 主要研究方向机器学习,社交网络和粒计算.
E-mail:zhaoshuzs2002@hotmail.com

张燕平　女,1962 年 2 月出生,安徽巢湖人. 安徽大学教授,博士研究生导师. 主要研究方向为机器学习和粒计算.
E-mail:zhangyp2@163.com