第47卷 第12期 计 算 机 工 程 2021年12月
Vol.47 No.12 Computer Engineering December 2021

·人工智能与模式识别· 文章编号：1000-3428(2021)12-0122-09 文献标志码：A 中图分类号：TP18

一种保留社区结构信息的网络嵌入算法

吕少卿[1,2,3]，赵雪莉[1,2]，张 潘[1,2]，任新成[3]

（1.西安邮电大学 通信与信息工程学院，西安 710121；2.陕西省信息通信网络及安全重点实验室，西安 710121；
3.陕西省能源大数据智能处理省市共建重点实验室，陕西 延安 716000）

摘 要：现有网络嵌入算法大多只保留网络的微观结构信息，忽略了网络中普遍存在的社区结构信息。为提高网络表示质量，提出一种保留社区结构信息的网络嵌入算法PCNE。通过最大化节点之间的一阶和二阶相似性，对网络的微观结构进行建模，同时通过分解可反映网络社区结构信息的社区结构嵌入矩阵，对网络的社区结构信息进行建模。将构建的2个模型融合到统一的联合非负矩阵分解框架中，结合相似度矩阵和社区隶属度矩阵得到融合社区结构信息的节点表示向量。在5个真实公开数据集上进行节点分类实验，结果表明，与DeepWalk、Node2vec、LINE算法相比，PCNE可使Micro-F1值提升0.96%~13.1%，验证了算法的有效性。

关键词：网络嵌入；社区结构；非负矩阵分解；网络表示学习；复杂网络

开放科学（资源服务）标志码（OSID）：

中文引用格式：吕少卿，赵雪莉，张潘，等.一种保留社区结构信息的网络嵌入算法[J].计算机工程，2021，47(12)：122-130.

英文引用格式：LÜ S Q，ZHAO X L，ZHANG P，et al.A network embedding algorithm Preserving community structure information[J].Computer Engineering，2021，47(12)：122-130.

A Network Embedding Algorithm Preserving Community Structure Information

LÜ Shaoqing[1,2,3]，ZHAO Xueli[1,2]，ZHANG Pan[1,2]，REN Xincheng[3]

（1. School of Communications and Information Engineering，Xi'an University of Posts and Telecommunications，Xi'an 710121，China；
2. Shaanxi Key Laboratory of Information Communication Network and Security，Xi'an 710121，China；
3. Shaanxi Key Laboratory of Intelligent Processing for Big Energy Data，Yan'an，Shaanxi 716000，China）

【Abstract】Most existing network embedding algorithms only retain the micro-structure information of the network，but ignore the community structure information which is important in networks. In order to incorporate the community structure information into the network embedding to improve the quality of network representation，a network embedding algorithm preserving community information in network embedding，named PCNE that preserves community structure information is proposed. The micro-structure of the network is modeled by maximizing the first-order and second-order similarity between nodes，and then the community structure information of the network is modeled by factorizing the community structure embedding matrix which can reflect the community structure information of the network.Under the joint supervision of the similarity matrix of the micro-structure and the community membership matrix of the meso-structure，PCNE obtains the node representation vectors that fused the community structure information by merging the both into a unified joint non-negative matrix factorization framework. The performance of the PCNE algorithm is evaluated by node classification experiments on five real public datasets and compared with other five state-of-the-art embedding models. Experimental results show that the proposed method improves the Micro-F1 by 0.96%~13.1% compared with classical algorithms such as DeepWalk，Node2Vec and LINE，thus verifying the effectiveness of PCNE.

【Key words】network embedding；community structure；non-negative matrix factorization；network representation learning；complex network

DOI：10. 19678/j. issn. 1000-3428. 0059448

基金项目：陕西省教育厅科研计划项目(17JK0703)；陕西省能源大数据智能处理省市共建重点实验室开放基金(IPBED10)；陕西省工业领域一般项目基金(2020GY-081)。

作者简介：吕少卿(1987—)，男，讲师、博士，主研方向为网络表示学习、数据挖掘；赵雪莉、张 潘，硕士研究生；任新成，教授。

收稿日期：2020-09-07 **修回日期：**2020-11-16 **E-mail：**lvsq3601@xupt.edu.cn

0　概述

复杂网络[1-3]作为一种特殊的数据类型在现实世界中随处可见，例如由社交平台上用户好友关系抽象得到的社交网络、由论文之间引用关系抽象得到的引文网络、由蛋白质之间的相互作用关系抽象得到的蛋白质网络、由网页间链接关系抽象得到的Web网络等。这些网络结构复杂，其中蕴含着一些值得深入探索和挖掘的信息及规律。

网络的表现形式很大程度上决定了能否对网络进行有效分析。早期，人们用邻接矩阵来进行网络的存储和表达，但邻接矩阵只能体现节点之间的链接关系，并不能体现网络的高阶关系[2]。此外，随着网络规模的增大，邻接矩阵还面临着存储成本高、计算效率低、数据稀疏等问题[4]。因此，研究人员转而研究将节点表示为低维、稠密的实值向量形式，即网络嵌入[5]（又称为网络表示学习）。将网络节点表示为低维、稠密向量就能够进行后续的网络分析任务，如节点分类[6]、链接预测[7]、社区发现[8]、可视化[9]等，还可以作为边信息应用到推荐系统[10]等其他任务中。

从算法所使用的工具角度，可将现有的网络嵌入算法分为基于矩阵特征向量的方法、基于矩阵分解的方法、基于简单神经网络的方法和基于深度神经网络的方法4类。基于矩阵特征向量的方法是早期的网络嵌入算法，包括局部线性嵌入[11]（Locally Linear Embedding，LLE）、拉普拉斯特征映射[12]（Laplacian Eigenmap，LE）等，该类算法通过提取网络的关系矩阵（如邻接矩阵或拉普拉斯矩阵），然后计算得到关系矩阵的特征向量，继而得到节点的表示向量。基于矩阵分解的方法包括GraRep[13]、HOPE[14]、NEU[15]等，该类算法通过对描述网络的关系矩阵进行矩阵分解，达到降维的目的，从而得到节点的低维表示向量。基于简单神经网络的方法包括DeepWalk[16]、Node2vec[17]和LINE[18]等，该类算法对网络进行概率建模，通过最大化概率来保留网络的拓扑结构信息，从而得到节点的表示向量。基于深度神经网络的方法包括SDNE[19]、DNGR[20]等，该类算法通过深度自编码捕获网络的非线性关系，进而得到节点的表示。

虽然上述方法保留了网络中的微观结构信息并取得了较好的表示结果，但却都忽略了网络结构中普遍存在的社区结构信息[1]。社区结构是网络所具有的宏观结构信息，同一社区内的节点通常具有密集的链接关系以及相似的特性，而不同社区节点间的链接则相对稀疏[21]。社区结构普遍存在于现实网络中，如社交网络、生物网络、Web网络、引文网络等[21-22]，其对刻画网络节点关系和完成后续网络分析任务具有重要作用。鉴于此，本文提出一种保留社区结构信息的网络嵌入算法PCNE。通过构造社区结构嵌入矩阵和社区隶属度矩阵得到原始网络中的宏观社区结构信息，并通过融合一阶相似性和二阶相似性得到网络中的微观结构信息。在此基础上，以迭代优化的方式对微观结构信息、宏观社区结构嵌入矩阵和社区隶属度矩阵进行联合优化，得到同时包含网络微观一阶、二阶结构信息和宏观社区结构信息的网络嵌入向量。

1　本文方法

表1列出了本文PCNE算法使用的符号及定义。

表1　符号定义

Table 1　Definition of symbols

符号	定义
n	网络中节点的数量
k	社区数量
d	节点的表示维数
$\boldsymbol{A}\in\mathbb{R}^{n\times n}$	网络的邻接矩阵
$\boldsymbol{S}\in\mathbb{R}^{n\times n}$	网络结构的二阶相似度矩阵
$\boldsymbol{P}\in\mathbb{R}^{n\times n}$	社区结构嵌入矩阵
$\boldsymbol{U}\in\mathbb{R}^{n\times k}$	社区隶属度矩阵
$\boldsymbol{H}\in\mathbb{R}^{k\times d}$	社区表示矩阵
$\boldsymbol{Y}\in\mathbb{R}^{n\times d}$	节点的表示矩阵

1.1　相关概念

本小节给出本文工作相关的一些基本概念及定义。

定义1（社区结构[23]）　社区结构是网络中存在的一些连接密集的群落（也称为簇）结构。同一社区内的节点彼此连接紧密，而各个不同社区中的节点间连接相对稀疏。

定义2（网络嵌入[5]）　网络嵌入又称网络表示学习或图嵌入。给定一个无向网络$G=(V,E)$，网络嵌入的目标是学习一个映射函数$f:v\rightarrow\boldsymbol{r}_v\in\mathbb{R}^d$，将网络中每一个节点映射为一个$d$维稠密的实数向量，并且满足$d<<|V|$。

1.2　算法框架

给定网络$G=(V,E)$。设$\boldsymbol{A}\in\mathbb{R}^{n\times n}$为网络的邻接矩阵。若节点$i$和节点$j$之间存在链接关系，则$\boldsymbol{A}$中对应元素为1，否则为0。所得到的节点的表示矩阵为$\boldsymbol{Y}$，$\boldsymbol{Y}$中的第i行代表节点i的表示向量，其中，d表示嵌入空间中节点表示向量的维数。

PCNE算法框架如图1所示，其中主要包含两部分内容，即保留网络微观结构信息的模型和保留社区结构信息的模型。具体而言，首先通过节点间的链接关系得到包含一阶相似性和二阶相似性的微观结构相似性矩阵$\boldsymbol{F}$，借助对称非负矩阵分解模型得到保留网络微观结构信息的损失函数；然后引入社区结构嵌入矩阵$\boldsymbol{P}$，通过联合非负矩阵分解模型得到保留网络社区结构信息的损失函数；最后通过联合优化两部分损失函数，得到同时保留网络微观结构信息和网络社区结构信息的节点表示。

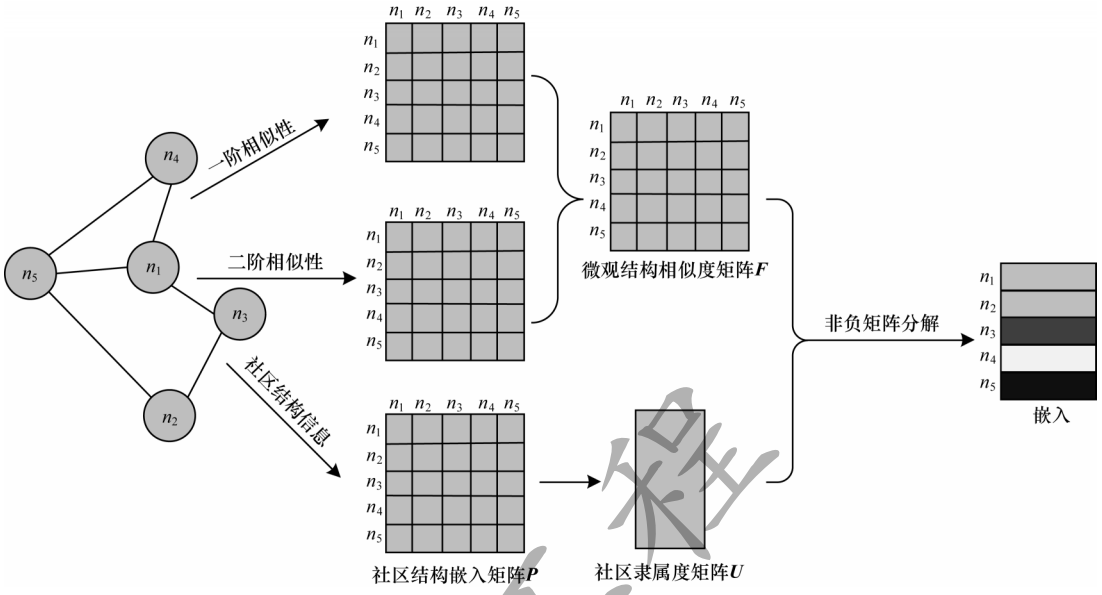

图1 PCNE模型框架

Fig.1 Framework of PCNE model

1.3 微观结构信息建模

本文通过保留网络中每对节点的一阶相似性和二阶相似性刻画网络的微观结构信息。具体地，如果在原始网络中一对节点之间存在边，那么它们就具有一阶相似性；如果一对节点在原始网络中没有边连接，那么它们之间的一阶相似性为0。在现实世界网络中，有边相连的两个节点之间通常具有相近的性质。以社交网络为例，如果一个节点和另一个节点相连（即有好友关系），那么这两个用户大概率具有相似的兴趣爱好或职业。这表明，在原始网络中直接相连的两个节点在嵌入空间中也应该彼此接近。本文将网络的邻接矩阵$\boldsymbol{A}$作为节点间网络结构一阶相似性的描述。

现实网络中存在的边通常是稀疏的[24]，没有连边的节点对并不代表它们在低维空间的表示向量不相似。一种补充一阶相似性缺陷的解决方案是考虑他们的共同邻居，通过共同邻居来衡量两个节点之间的相似性，即二阶相似性。此外，现实中的网络经常会由于观测手段的不足导致丢失一些网络中本该存在的链接关系，因此保留节点间的二阶相似性就更有必要。本文定义矩阵$\boldsymbol{S}\in\mathbb{R}^{n\times n}$为二阶相似度矩阵，并用邻接矩阵行向量的余弦相似作为其二阶相似度，如式(1)所示：

$$s_{ij}=\frac{\boldsymbol{a}_i\boldsymbol{a}_j}{\|\boldsymbol{a}_i\|\|\boldsymbol{a}_j\|} \tag{1}$$

其中：$\boldsymbol{a}_i$为邻接矩阵$\boldsymbol{A}$的第i行。

为同时保留网络结构的一阶相似性和二阶相似性，本文将两种相似性的加权和作为网络最终的微观结构相似性，用矩阵$\boldsymbol{F}$来表示，并命名为微观结构相似度矩阵。$\boldsymbol{F}$计算公式如式(2)所示：

$$\boldsymbol{F}=\boldsymbol{A}+\alpha\boldsymbol{S} \tag{2}$$

其中：参数$\alpha>0$，为二阶相似性的权重系数，α的大小体现二阶相似性对节点表示的重要性。在本文的后续实验中，选择$\alpha=3$。

由于本文针对的是无向网络，因此微观结构相似度矩阵$\boldsymbol{F}$是一个非负的对称矩阵。为使所得到的节点表示能够保留网络的微观结构信息，本文基于对称非负矩阵分解[23]提出式(3)所示的损失函数来最小化节点之间的嵌入差异：

$$\min\|\boldsymbol{F}-\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\|_{\mathrm{F}}^2 \tag{3}$$

其中：符号$\|\cdot\|_{\mathrm{F}}$表示矩阵的Frobenius范数（简称F范数）。

1.4 社区结构信息建模

网络的邻接矩阵反映的是网络节点之间的链接关系，文献[23]通过非负矩阵分解的方式提取网络中的社区结构信息。假设网络由k个社区组成，则其目标函数如式(4)所示：

$$\begin{aligned}&\min\|\boldsymbol{A}-\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\|_{\mathrm{F}}^2\\&\text{s.t. }\boldsymbol{U}\geqslant\boldsymbol{0}\end{aligned} \tag{4}$$

其中：矩阵$\boldsymbol{A}$为网络的邻接矩阵；$\boldsymbol{U}\in\mathbb{R}^{n\times k}$为社区隶属度矩阵，反映网络中各节点与各个社区之间的从属关系。无论节点i和节点j是否属于同一个社区，只要两节点之间存在链接，其对应元素就为1，因此，蕴含在网络中的社区结构不能通过直接分解邻接矩阵$\boldsymbol{A}$来反映。本文通过分解文献[25]中所提出的更能反映网络中社区结构信息的社区结构嵌入矩阵$\boldsymbol{P}$来提取网络的社区结构分布信息，其目标函数形式化表示为：

$$\begin{aligned}&\min J_{\mathrm{C}}=\|\boldsymbol{P}-\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\|_{\mathrm{F}}^2\\&\text{s.t. }\boldsymbol{U}\geqslant\boldsymbol{0}\end{aligned} \tag{5}$$

下面介绍社区结构嵌入矩阵$\boldsymbol{P}$的具体构造方法。

由于社区内部的节点彼此之间连接紧密，因此每个具有链接关系的节点对都有落入同一社区内的可能性，定义这种可能性为社区成员相似性。对于

网络中的节点i和节点j,它们之间社区成员相似度的计算公式如式(6)所示:

$$f(i,j)=2\sigma(\boldsymbol{u}_i\boldsymbol{u}_j^{\mathrm{T}})-1=\frac{2}{1+\mathrm{e}^{-\boldsymbol{u}_i\boldsymbol{u}_j^{\mathrm{T}}}}-1 \tag{6}$$

现实世界中的大多数网络都比较稀疏,在网络中任意选择两个节点,他们之间存在边的可能性几乎为零。为了最大化具有链接关系的节点对的社区成员相似性,同时最小化随机采样的节点对之间的社区成员相似性,本文采用负采样的方式,设计如式(7)所示的损失函数:

$$l(i,j)=\boldsymbol{A}_{ij}(\ln\sigma(\boldsymbol{u}_i\boldsymbol{u}_j^{\mathrm{T}})+n_sE_{j_{\mathrm{N}}\sim P_{\mathrm{V}}}[\ln\sigma(-\boldsymbol{u}_i\boldsymbol{u}_{j_{\mathrm{N}}}^{\mathrm{T}})]) \tag{7}$$

其中:n_s为负采样样本数;j_{N}为负采样的随机采样节点;$P_{\mathrm{V}}(i)=\dfrac{d_i}{D}$,$d_i$表示节点$i$的度,$D$表示整个网络节点度数的总和。式(7)又可表示为:

$$l(i,j)=\boldsymbol{A}_{ij}\ln\sigma(\boldsymbol{u}_i\boldsymbol{u}_j^{\mathrm{T}})+n_s\frac{d_id_j}{D}\ln\sigma(-\boldsymbol{u}_i\boldsymbol{u}_j^{\mathrm{T}}) \tag{8}$$

对式(8)求偏导,得到:

$$\frac{\partial l(i,j)}{\partial(\boldsymbol{u}_i\boldsymbol{u}_j^{\mathrm{T}})}=\boldsymbol{A}_{ij}\sigma(-\boldsymbol{u}_i\boldsymbol{u}_j^{\mathrm{T}})-n_s\frac{d_id_j}{D}\sigma(\boldsymbol{u}_i\boldsymbol{u}_j^{\mathrm{T}}) \tag{9}$$

令偏导$\dfrac{\partial l(i,j)}{\partial(\boldsymbol{u}_i\boldsymbol{u}_j^{\mathrm{T}})}=0$,可得:

$$\boldsymbol{u}_i\boldsymbol{u}_j^{\mathrm{T}}=\ln\frac{\boldsymbol{A}_{ij}D}{n_sd_id_j} \tag{10}$$

基于以上分析,社区结构嵌入矩阵$\boldsymbol{P}$可由式(11)进行构造:

$$p_{ij}=\max\left\{\ln\frac{\boldsymbol{A}_{ij}D}{n_sd_id_j},0\right\} \tag{11}$$

1.5 保留社区结构信息的网络嵌入

为了将网络的社区结构信息融入网络嵌入的过程中,利用得到的社区隶属度矩阵$\boldsymbol{U}$对网络表示学习进行约束,使所得节点表示能够反映出网络的社区结构信息,从而在一定程度上提高网络表示学习的质量。本文引入一个非负矩阵$\boldsymbol{H}\in\mathbb{R}^{k\times d}$,并命名为社区表示矩阵,其第$i$行的行向量$\boldsymbol{h}_i$即为第$i$个社区的向量表示。如果某节点的表示向量与某一社区的表示向量相似,则认为该节点属于该社区的可能性高。本文将节点i的表示向量$\boldsymbol{y}_i$与社区r的表示向量$\boldsymbol{h}_r$的内积$\boldsymbol{y}_i\boldsymbol{h}_r^{\mathrm{T}}$作为两向量之间相似性的表达。因此,若节点$i$的表示向量与社区$r$的表示向量相互正交,即两者的表示完全不同,那么节点i属于社区r的可能性近乎为零。由于社区隶属度矩阵$\boldsymbol{U}$体现了每个节点与各社区之间的从属信息,因此本文希望$\boldsymbol{Y}\boldsymbol{H}^{\mathrm{T}}$的结果与社区隶属度$\boldsymbol{U}$尽可能一致,从而设计如式(12)所示的目标函数:

$$\min\|\boldsymbol{U}-\boldsymbol{Y}\boldsymbol{H}^{\mathrm{T}}\|_{\mathrm{F}}^2 \tag{12}$$

基于上述分析,本文在网络的微观结构模型和社区发现模型之间建立了纽带,进而挖掘网络表示学习的过程中的社区结构信息。最终目标函数如式(13)所示:

$$\min J=\|\boldsymbol{F}-\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\|_{\mathrm{F}}^2+\beta\|\boldsymbol{P}-\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\|_{\mathrm{F}}^2+\gamma\|\boldsymbol{U}-\boldsymbol{Y}\boldsymbol{H}^{\mathrm{T}}\|_{\mathrm{F}}^2 \tag{13}$$

其中:参数β和γ均为非负值,β用于提取网络中蕴藏的社区结构,γ用于调节社区结构信息在网络表示学习过程中的贡献大小。通过调节这两个参数的值可以适应不同的网络和不同的应用场景。

1.6 模型优化

由于式(13)所示的目标函数是非凸的,因此几乎不可能找到全局最优解。为解决该最优化问题,本文基于文献[26]提出一个能够保证收敛到局部最优解的迭代更新过程。在保持其他参数不变的情况下,迭代更新每一个参数,具体过程如下:

更新$\boldsymbol{H}$:保持$\boldsymbol{Y}$、$\boldsymbol{U}$不变,式(13)中只有最后一项与矩阵$\boldsymbol{H}$有关。由文献[27]所提非负矩阵分解模型的乘法更新规则,得到式(14):

$$h_{ij}\leftarrow h_{ij}\frac{(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{Y})_{ij}}{(\boldsymbol{H}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})_{ij}} \tag{14}$$

更新$\boldsymbol{U}$:保持$\boldsymbol{Y}$、$\boldsymbol{H}$不变,为更新矩阵$\boldsymbol{U}$,本文需要解决一个联合矩阵分解问题。由于式(13)中只有最后两项与矩阵$\boldsymbol{U}$有关,因此只需要最小化式(15)所示的损失函数:

$$\min_{\boldsymbol{U}\geqslant\boldsymbol{0}}J(\boldsymbol{U})=\beta\|\boldsymbol{P}-\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\|_{\mathrm{F}}^2+\gamma\|\boldsymbol{U}-\boldsymbol{Y}\boldsymbol{H}^{\mathrm{T}}\|_{\mathrm{F}}^2 \tag{15}$$

由矩阵的F范数和迹(trace)之间的关系$\|\boldsymbol{A}\|_{\mathrm{F}}^2=\mathrm{tr}(\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A})$以及矩阵迹运算法可得式(16):

$$\min_{\boldsymbol{U}\geqslant\boldsymbol{0}}J(\boldsymbol{U})=\beta\mathrm{tr}(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}-2\boldsymbol{P}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}+\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}})+\gamma\mathrm{tr}(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}-2\boldsymbol{U}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{H}^{\mathrm{T}}+\boldsymbol{H}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{H}^{\mathrm{T}}) \tag{16}$$

该约束优化问题可以通过引入矩阵$\boldsymbol{U}$的拉格朗日乘数矩阵$\boldsymbol{\Theta}$构造拉格朗日函数来解决。拉格朗日函数如式(17)所示:

$$L(\boldsymbol{U})=\beta\|\boldsymbol{P}-\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\|_{\mathrm{F}}^2+\gamma\|\boldsymbol{U}-\boldsymbol{Y}\boldsymbol{H}^{\mathrm{T}}\|_{\mathrm{F}}^2-\mathrm{tr}(\boldsymbol{\Theta}\boldsymbol{U}^{\mathrm{T}}) \tag{17}$$

对矩阵$\boldsymbol{U}$求偏导,得到:

$$\frac{\partial L(\boldsymbol{U})}{\partial\boldsymbol{U}}=\beta(4\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}-4\boldsymbol{P}^{\mathrm{T}}\boldsymbol{U})+\gamma(2\boldsymbol{U}-2\boldsymbol{Y}\boldsymbol{H}^{\mathrm{T}})-\boldsymbol{\Theta} \tag{18}$$

令$\dfrac{\partial L(\boldsymbol{U})}{\partial\boldsymbol{U}}=0$,同时引入Karush-Kuhn-Tucker(KKT)松弛互补条件$\Theta_{ij}U_{ij}=0$,可得如下更新规则:

$$u_{ij}\leftarrow u_{ij}\left(\frac{(2\beta\boldsymbol{P}^{\mathrm{T}}\boldsymbol{U}+\gamma\boldsymbol{Y}\boldsymbol{H}^{\mathrm{T}})_{ij}}{(2\beta\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}+\gamma\boldsymbol{U})_{ij}}\right)^{\frac{1}{4}} \tag{19}$$

同理,固定矩阵$\boldsymbol{U}$和$\boldsymbol{H}$,可得到矩阵$\boldsymbol{Y}$的更新规则,如式(20)所示:

$$y_{ij}\leftarrow y_{ij}\left(\frac{(2\boldsymbol{F}\boldsymbol{Y}+\gamma\boldsymbol{U}\boldsymbol{H})_{ij}}{(2\boldsymbol{Y}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y}+\gamma\boldsymbol{Y}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{H})_{ij}}\right)^{\frac{1}{4}} \tag{20}$$

上述更新规则均具有收敛性,通过迭代交替更新矩阵$\boldsymbol{U}$、$\boldsymbol{H}$、$\boldsymbol{Y}$,可以得到网络的节点表示矩阵$\boldsymbol{Y}$。

PCNE算法的伪代码如下所示:

算法1 PCNE算法

输入 属性网络$G=(V,E)$,迭代次数i,参数β和γ,负采样样本数n_s,表示向量维度d

输出 网络节点表示矩阵$\boldsymbol{Y}$

1.计算邻接矩阵A
2.根据式(1)计算二阶相似度矩阵S
3.根据式(2)计算微观结构相似度矩阵F
4.根据式(11)计算社区结构嵌入矩阵P
5.初始化矩阵H、U和Y
6.J=0
7.for iter=1 to i
8.根据式(13)计算损失函数J
9.if 相邻两次损失函数J的差值小于阈值
10.end for
11.else
12.根据式(14)更新H
13.根据式(19)更新U
14.根据式(20)更新Y
15.end
16.return Y

1.7 复杂度分析

PCNE算法的时间复杂度主要取决于式(14)、式(19)和式(20)的矩阵乘法运算,它们的时间复杂度分别为$O(ndk)$、$O(n^2k+ndk)$和$O(n^2d+ndk)$,其中:n为节点数量;d为节点表示维数;k为网络中的社区数。由于实际应用中满足$k\ll n$,因此PCNE的算法复杂度为$O(dn^2)$。DeepWalk的算法复杂度为$O(dn\log_a n)$[16],Node2vec的算法复杂度为$O(dn)$[17],算法LINE的复杂度为$O(dm)$[18],其中m为网络中边的数量。虽然这些算法的复杂度比PCNE算法低,但是DeepWalk和Node2vec都是基于随机游走的算法,只考虑了节点的局部链接关系,LINE只考虑了一阶、二阶信息,这些算法都没有考虑宏观的社区结构信息,而社区结构信息对现实网络分析非常重要。PCNE通过非负矩阵分解的方式将社区结构信息融入到网络表示学习过程中,在后续网络分析任务中能够取得比DeepWalk、Node2vec、LINE更好的效果。

2 实验与结果分析

2.1 实验数据集

本文实验选取公开的真实网络Karate[28]和WebKB网络[29]的4个子网络(Cornell、Texas、Washington、Wisconsin)作为数据集进行实验,如表2所示。Karate[30]数据集是描述美国一所大学空手道俱乐部中34个成员之间社会关系的网络,由34个节点和78条边组成,包含2个类别标签;Cornell、Texas、Washington、Wisconsin是WebKB[31]数据集的4个子网络,均包含5个类别标签,分别是由美国4所大学的网页之间的链接关系构建的,Cornell数据集由195个节点和283条边组成,Texas数据集由187个节点和289条边组成,Washington数据集由230个节点和366条边组成,Wisconsin数据集由265个节点和469条边组成。

表2 实验数据集信息

Table 2 Information of datasets for experiment

数据集	节点数量	连边数量	类别
Karate	34	78	2
Cornell	195	283	5
Texas	187	289	5
Washington	230	366	5
Wisconsin	265	469	5

2.2 对比算法与参数设置

为验证本文PCNE算法的有效性,将其与以下5个具有代表性的网络表示学习算法进行对比。

1)DeepWalk[16]算法。该算法通过随机游走模型将获取的节点序列看作文本中的单词,作为Word2vec算法的输入,通过Skip-Gram模型训练得到各节点的表示。实验中参数设置:每个节点采样的序列数量为40,节点序列长度为40,窗口大小为10。

2)LINE[18]算法。实验中用LINE1和LINE2分别表示基于一阶结构相似性的模型和基于二阶结构相似性的模型,用LINE表示基于一阶和二阶结构相似性的模型。这3个算法的参数设置为:负采样的样本数设为5,学习率的初值设为0.025。

3)Node2vec[17]算法。该算法在DeepWalk算法的基础上引入2个超参数p、q以平衡基于深度的随机游走策略和基于广度的随机游走策略。Node2vec算法的参数设置为$p=0.25,q=0.25$,其余参数与DeepWalk的参数一致。

为保证实验的公平性,节点的表示向量维度都设置为20维。本文PCNE算法的参数设置为:$\beta=0.1,\gamma=0.5$。

2.3 实验结果及其分析

本文利用节点分类任务评估PCNE算法的性能。为了执行该任务,将所得到的节点嵌入向量视为网络中每个节点的特征作为分类器的输入,以预测节点的标签。在实验中,本文采用scikit-learn包中的一对多SVM分类器,在分类器的训练过程中,训练集百分比即训练率A设置为{10%,15%,20%,25%,30%},其余部分作为测试集。为确保实验结果的稳定性和可靠性,对每个数据集分别进行10次独立重复实验,最终实验结果取10次实验的Micro-F1值和Macro-F1值的均值。表3~表7列出了PCNE和其他5个算法在5个数据集不同训练比例下的实验结果,其中加粗数据表示最优。可以看出,在Karate、Texas和Washington数据集上,PCNE算法明显优于其他算法,无论是以Micro-F1还是以Macro-F1为评价标准,其在各训练比例下均取得了最高的评价得分:节点分类性能在Micro-F1上分别比第2名提高了0.96%~9.36%(Karate)、0.77%~3.51%

（Texas）和9.95%~13.01%（Washington），在Macro-F1上分别比第2名提高了3.02%~11.44%（Karate）、0.4%~5.66%（Texas）和3.82%~5.93%（Washington）。在Cornell和Wisconsin数据集上，PCNE虽然没有在所有数据上体现出优势，但在大部分情况下的表现依然具有一定竞争力。例如：在Cornell数据集上，PCNE在节点分类任务上得到了最高的Micro-F1，虽在Macro-F1分数表现略差，但比最高得分仅低了1%左右；而在Wisconsin数据集上，其节点分类性能指标Micro-F1也在各训练率下均取得了最高评价得分。因此，从综合性能上看，PCNE算法在节点分类任务中的表现仍具有较强的竞争力。

表3　Karate数据集上的节点分类性能

Table 3　Node classification performance on Karate dataset　%

算法	Micro-F1值					Macro-F1值				
	10%训练率	15%训练率	20%训练率	25%训练率	30%训练率	10%训练率	15%训练率	20%训练率	25%训练率	30%训练率
DeepWalk	62.90	75.51	82.14	89.61	94.16	55.74	71.84	79.59	88.71	94.13
LINE1	48.71	46.55	46.43	46.53	45.83	34.28	35.08	38.17	35.49	36.59
LINE2	46.45	46.55	43.93	42.31	43.33	31.68	34.24	33.14	33.61	32.65
LINE	47.10	49.66	49.29	51.15	52.92	33.40	40.16	42.90	45.54	47.75
Node2vec	67.10	81.38	80.00	85.38	87.92	59.19	79.68	78.19	84.51	87.38
PCNE	**68.06**	**84.62**	**85.50**	**93.15**	**97.25**	**62.63**	**82.70**	**83.93**	**92.83**	**97.22**

表4　Cornell数据集上的节点分类性能

Table 4　Node classification performance on Cornell dataset　%

算法	Micro-F1值					Macro-F1值				
	10%训练率	15%训练率	20%训练率	25%训练率	30%训练率	10%训练率	15%训练率	20%训练率	25%训练率	30%训练率
DeepWalk	35.40	36.45	36.67	36.46	38.03	19.47	19.60	**20.64**	**21.06**	**21.75**
LINE1	32.44	32.83	32.12	32.99	33.72	16.28	17.06	16.19	15.80	15.60
LINE2	32.27	33.86	35.00	34.29	35.47	17.15	16.04	15.78	15.31	15.72
LINE	35.34	35.06	35.83	36.05	37.59	14.66	14.24	15.41	16.56	17.32
Node2vec	34.26	35.12	33.85	33.95	33.43	18.64	**20.73**	19.56	20.31	20.03
PCNE	**36.97**	**38.22**	**39.00**	**39.22**	**39.78**	**19.71**	20.24	19.28	20.50	20.84

表5　Texas数据集上的节点分类性能

Table 5　Node classification performance on Texas dataset　%

算法	Micro-F1值					Macro-F1值				
	10%训练率	15%训练率	20%训练率	25%训练率	30%训练率	10%训练率	15%训练率	20%训练率	25%训练率	30%训练率
DeepWalk	49.41	47.92	48.87	48.37	50.15	22.31	22.11	22.63	20.74	22.88
LINE1	52.66	53.21	52.87	52.76	52.44	14.79	15.46	15.20	15.15	14.75
LINE2	51.48	53.65	52.00	53.33	54.27	14.76	15.35	15.80	15.40	14.98
LINE	49.05	51.32	54.53	54.96	54.89	14.08	15.33	15.92	15.61	15.95
Node2vec	48.22	48.24	47.73	48.01	49.31	19.86	19.78	20.44	21.23	22.23
PCNE	**53.43**	**55.66**	**56.20**	**57.52**	**58.04**	**22.71**	**24.27**	**24.16**	**26.89**	**27.49**

表6　Washington数据集上的节点分类性能

Table 6　Node classification performance on Washington dataset　%

算法	Micro-F1值					Macro-F1值				
	10%训练率	15%训练率	20%训练率	25%训练率	30%训练率	10%训练率	15%训练率	20%训练率	25%训练率	30%训练率
DeepWalk	40.53	40.46	42.34	43.99	45.90	19.76	19.53	21.31	21.68	21.39
LINE1	41.69	42.40	43.20	42.89	42.73	14.74	14.09	14.74	14.09	14.69
LINE2	40.00	40.05	40.76	43.06	43.04	17.68	17.15	16.32	17.31	16.57
LINE	41.98	42.40	42.61	43.18	44.10	13.56	14.15	14.63	14.02	15.92
Node2vec	40.97	43.37	44.89	44.62	46.77	21.25	21.19	21.79	21.29	22.63
PCNE	**52.32**	**53.32**	**56.47**	**57.63**	**59.32**	**25.08**	**25.01**	**27.72**	**27.58**	**27.80**

表7 Wisconsin数据集上的节点分类性能

Table 7 Node classification performance on Wisconsin dataset %

算法	Micro-F1值					Macro-F1值				
	10%训练率	15%训练率	20%训练率	25%训练率	30%训练率	10%训练率	15%训练率	20%训练率	25%训练率	30%训练率
DeepWalk	42.01	42.39	42.69	42.36	42.04	24.83	24.89	27.11	25.55	25.75
LINE1	38.66	41.15	40.99	41.76	41.51	15.16	15.78	15.01	15.37	16.00
LINE2	39.79	40.49	40.90	41.26	41.88	13.66	14.14	15.38	16.14	15.86
LINE	39.92	39.60	41.23	40.10	40.97	15.45	15.11	13.96	14.99	14.33
Node2vec	41.13	38.94	40.33	42.21	42.42	24.56	24.90	25.70	26.20	24.71
PCNE	**47.70**	**46.02**	**47.78**	**48.54**	**49.03**	19.49	20.69	21.45	22.37	22.76

2.4 参数敏感性分析

本文PCNE算法主要包含β、γ、d这3个参数，分别用以调节各目标对网络表示学习的贡献大小，其中：参数β控制网络中社区结构划分的质量对网络表示学习的影响；参数γ控制社区结构信息对网络表示学习的影响；参数d为所学节点向量表示维数。为研究这3个参数对PCNE算法的影响，分别在Cornell、Texas、Washington和Wisconsin数据集上进行实验分析。

1)在固定向量表示维数d=100的情况下，对其余2个参数的敏感性进行实验分析。实验中将训练比例固定为30%，并设置参数$\beta,\gamma\in\{0.1,0.5,1,5,10\}$。图2~图5分别记录并反映了不同数据集上2个参数对节点分类任务评价指标Micro-F1值的影响，可以看出：在4个数据集上，随着参数β和γ的调整，PCNE算法表现都较为稳定，性能指标Micro-F1在4个数据集上的波动范围均在可控范围内，分别为2.89%(Cornell)、4.65%(Texas)、3.17%(Washington)和3.36%(Wisconsin)；而从整体看，随着2个参数的变化，指标Micro-F1变化范围较小，变化趋势较为平缓。

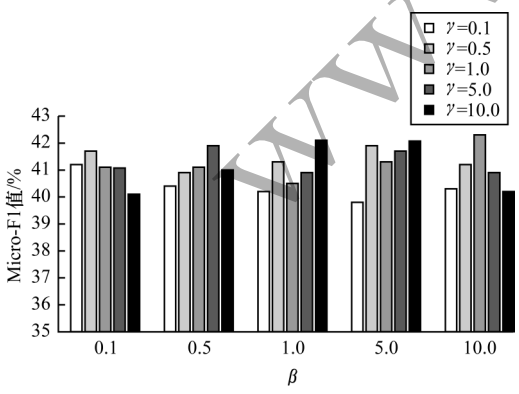

图2 Cornell数据集上参数β和γ的敏感性

Fig.2 Susceptibility of parameters β and γ on Cornell dataset

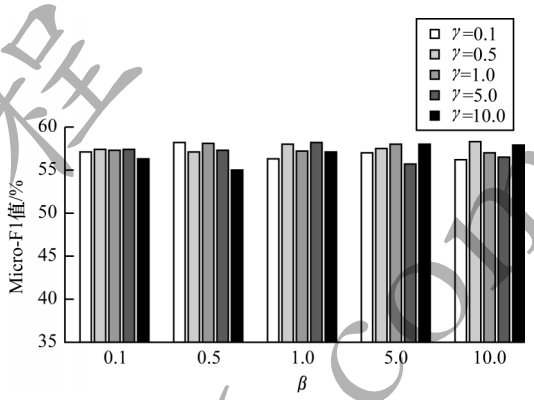

图3 Texas数据集上参数β和γ的敏感性

Fig.3 Susceptibility of parameters β and γ on Texas dataset

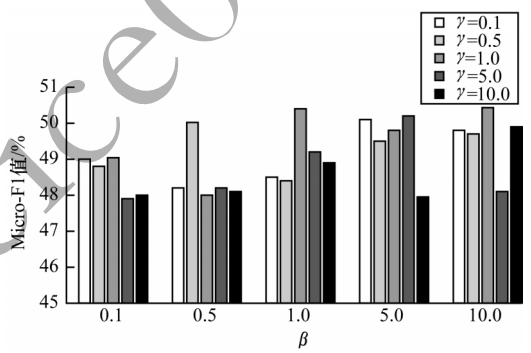

图4 Washington数据集上参数β和γ的敏感性

Fig.4 Susceptibility of parameters β and γ on Washington dataset

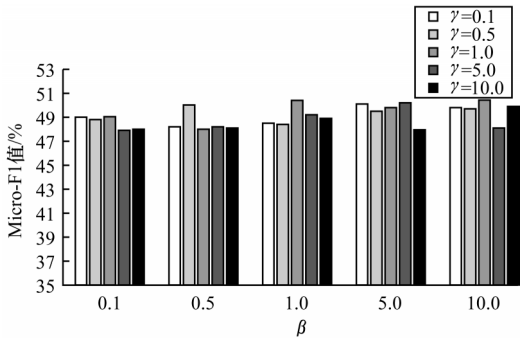

图5 Wisconsin数据集上参数β和γ的敏感性

Fig.5 Susceptibility of parameters β and γ on Wisconsin dataset

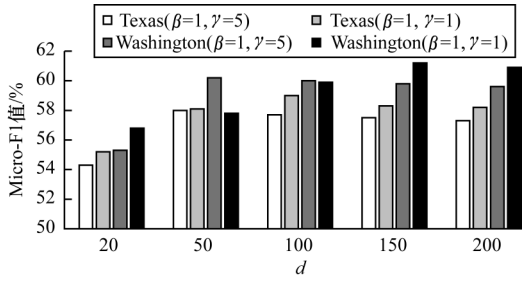

图6　表示向量维数 d 的敏感性

Fig.6　Susceptibility of dimension d for representation vector

2）在固定参数 β 和 γ 的情况下，分析表示向量维数 d 的敏感性，即 d 对节点分类性能的影响。实验中训练比例仍固定为30%，并设置参数 $d\in\{20,50,100,150,200\}$。由于在其他数据集上也会出现相似的结果，因此本文仅展示在数据集Texas和Washington上的结果。图6记录并反映了在这2个数据集上表示维数 d 对于分类性能评价指标Micro-F1的影响，可以看出：无论是Texas数据集还是Washington数据集，随着表示维数 d 的增加，节点分类性能指标Micro-F1也随之增大，当 d 增加到一定数值后，Micro-F1便趋于稳定，再随着 d 增大，Micro-F1反而有所下降。这说明在表示维数较低的情况下，该算法捕获的网络信息较少，随着表示维数的增加，算法捕获网络信息的能力有所提高，而选择太高的维数又会因特征过多导致具有重要区分度的特征的权重过小从而影响节点间的差异。因此，对于Texas和Washington数据集而言，节点表示维数选取 d=100或50较为合适。

3　结束语

本文提出一种保留社区结构信息的网络表示学习算法PCNE。定义网络结构的一阶相似性和二阶相似性，将其作为网络的微观结构信息进行建模。同时对网络中蕴含的社区结构信息进行建模，通过非负矩阵分解的方式得到社区隶属度矩阵，基于此提取网络中的社区结构信息。将两者放在一个统一的框架下进行联合优化，从而得到保留社区结构信息及微观结构信息的节点表示向量。在真实数据集上的节点分类对比实验结果验证了PCNE算法的有效性。该算法与DeepWalk、Node2vec、LINE等算法都是针对同质网络进行研究（即网络中的节点为同一类型），虽然同质网络在现实世界中更广泛和普遍，但现实世界中还存在大量的异质网络（即在同一个网络中存在多种类型的节点，如评价网络、购物网络等）。因此，如何合理高效地将网络结构信息融入异质网络的表示学习，将是后续深入研究的课题。

参考文献

[1] PENG C, XIAO W, JIAN P, et al. A survey on network embedding[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 31(5): 833-852.

[2] BENSON A R, GLEICH D F, LESKOVEC J. Higher-order organization of complex networks[J]. Science, 2016, 353(6295): 163-166.

[3] CAI H, ZHENG V W, CHANG K C C. A comprehensive survey of graph embedding: problems, techniques, and applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2018, 30(9): 1616-1637.

[4] 涂存超，杨 成，刘知远，等. 网络表示学习综述[J]. 中国科学：信息科学，2017，47(8)：980-996.
TU C, YANG C, LIU Z, et al. Network representation learning: an overview[J]. SCIENTIA SINCA Informationic, 2017, 47(8): 980-996. (in Chinese)

[5] 陈维政，张岩，李晓明. 网络表示学习[J]. 大数据，2015，1(3)：8-22.
CHEN W, ZHANG Y, LI X. Network representation learning[J]. Big Data, 2015, 1(3): 8-22. (in Chinese)

[6] BHAGAT S, CORMODE G, MUTHUKRISHNAN S. Node classification in social networks[M]. Berlin, Germany: Springer, 2011.

[7] LÜ L, ZHOU T. Link prediction in complex networks: a survey[J]. Physica A: Statistical Mechanics and Its Applications, 2011, 390(6): 1150-1170.

[8] FORTUNATO S. Community detection in graphs[J]. Physics Reports, 2010, 486(3/4/5): 75-174.

[9] MAATEN L, HINTON G. Visualizing data using t-SNE[J]. Journal of Machine Learning Research, 2008, 9: 2579-2605.

[10] CHEN J, WU Y, FAN L, et al. N2VSCDNNR: a local recommender system based on Node2vec and rich information network[J]. IEEE Transactions on Computational Social Systems, 2019, 6(3): 456-466.

[11] ROWEIS S T, SAUL L K. Nonlinear dimensionality reduction by locally linear embedding[J]. Science, 2000, 290(5500): 2323-2326.

[12] BELKIN M, NIYOGI P. Laplacian eigenmaps and spectral techniques for embedding and clustering[C]//Proceedings of Advances in Neural Information Processing Systems. Cambridge, USA: MIT Press, 2001: 585-591.

[13] CAO S, LU W, XU Q. GraRep: learning graph representations with global structural information[C]//Proceedings of the 24th ACM International Conference on Information and Knowledge Management. New York, USA: ACM Press, 2015: 891-900.

[14] OU M, CUI P, PEI J, et al. Asymmetric transitivity preserving graph embedding[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2016: 1105-1114.

[15] YANG C, SUN M, LIU Z, et al. Fast network embedding enhancement via high order proximity approximation[C]// Proceedings of International Joint Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 2017: 3894-3900.

[16] 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. New York, USA: ACM Press, 2014: 701-710.

[17] GROVER A, LESKOVEC J. Node2vec: scalable feature learning for networks[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2016: 855-864.

[18] 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. New York, USA: ACM Press, 2015: 1067-1077.

[19] WANG D, CUI P, ZHU W. Structural deep network embedding[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press, 2016: 1225-1234.

[20] CAO S, LU W, XU Q. Deep neural networks for learning graph representations[C]//Proceedings of the 30th AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 2016: 1145-1152.

[21] GIRVAN M, NEWMAN M E J. Community structure in social and biological networks[J]. PNAS, 2001, 99: 7821-7826.

[22] CHERIFI H, PALLA G, SZYMANSKI B, et al. On community structure in complex networks: challenges and opportunities[J]. Applied Network Science, 2019, 4: 1-5.

[23] NEWMAN M E J. Finding community structure in networks using the eigenvectors of matrices[J]. Physical Review E, 2006, 74(3): 1-5.

[24] TU K, CUI P, WANG X, et al. Structural deep embedding for hyper-networks[C]//Proceedings of the 32nd AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 2018: 426-433.

[25] LI Y, SHA C, HUANG X, et al. Community detection in attributed graphs: an embedding approach[C]//Proceedings of the 32nd AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 2018: 1-5.

[26] LEVY O, GOLDBERG Y. Neural word embedding as implicit matrix factorization[C]//Proceedings of the 27th International Conference on Neural Information Processing Systems. New York, USA: ACM Press, 2014: 2177-2185.

[27] LEE D D, SEUNG H S. Algorithms for non-negative matrix factorization[C]//Proceedings of the 13th International Conference on Neural Information Processing Systems. New York, USA: ACM Press, 2000: 535-541.

[28] Karate[EB/OL]. [2020-08-10]. http://networkrepository.com/soc-karate.php.

[29] WebKB[EB/OL]. [2020-08-10]. http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/.

[30] ZACHARY W. An information flow model for conflict and fission in small groups[J]. Journal of Anthropological Research, 1977, 33(4): 452-473.

[31] CRAVEN M, DIPASQUO D, FREITAG D, et al. Learning to extract symbolic knowledge from the World Wide Web[C]// Proceedings of the 15th AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI Press, 1998: 1134-1142.

编辑 金胡考