DOI: 10.11992/tis.201905045
网络出版地址: http://kns.cnki.net/kcms/detail/23.1538.TP.20190902.1139.006.html

基于多粒度结构的网络表示学习

张蕾[1,2], 钱峰[1,2], 赵姝[1], 陈洁[1], 张燕平[1], 刘峰[1]
（1. 安徽大学 计算机科学与技术学院, 安徽 合肥 230601; 2. 铜陵学院 数学与计算机学院, 安徽 铜陵 244061）

摘　要: 图卷积网络 (GCN) 能够适应不同结构的图, 但多数基于 GCN 的方法难以有效地捕获网络的高阶相似性。简单添加卷积层将导致输出特征过度平滑并使它们难以区分, 而且深层神经网络更难训练。本文选择将网络的多粒度结构和图卷积网络结合起来用于学习网络的节点特征表示, 提出基于多粒度结构的网络表示学习方法 Multi-GS。首先, 基于模块度聚类和粒计算思想, 用分层递阶的多粒度空间替代原始的单层网络拓扑空间; 然后, 利用 GCN 模型学习不同粗细粒度空间中粒的表示; 最后, 由粗到细将不同粒的表示组合为原始空间中节点的表示。实验结果表明: Multi-GS 能够捕获多种结构信息, 包括一阶和二阶相似性、社团内相似性 (高阶结构) 和社团间相似性 (全局结构)。在绝大多数情况下, 使用多粒度的结构可改善节点分类任务的分类效果。

关键词: 网络表示学习; 网络拓扑; 模块度增量; 网络粒化; 多粒度结构; 图卷积网络; 节点分类; 链接预测
中图分类号: TN929.12　　**文献标志码:** A　　**文章编号:** 1673−4785(2019)06−1233−10

中文引用格式: 张蕾, 钱峰, 赵姝, 等. 基于多粒度结构的网络表示学习 [J]. 智能系统学报, 2019, 14(6): 1233–1242.
英文引用格式: ZHANG Lei, QIAN Feng, ZHAO Shu, et al. Network representation learning based on multi-granularity structure[J]. CAAI transactions on intelligent systems, 2019, 14(6): 1233–1242.

Network representation learning based on multi-granularity structure

ZHANG Lei[1,2], QIAN Feng[1,2], ZHAO Shu[1], CHEN Jie[1], ZHANG Yanping[1], LIU Feng[1]
(1. School of Computer Science and Technology, Anhui University, Hefei 230601, China; 2. School of Mathematics and Computer Science, Tongling University, Tongling 244061, China)

Abstract: The Graph Convolution Network (GCN) can adapt to graphs with different structures. However, most GCN-based models have difficulty effectively capturing the high-order similarity of the network. Simply adding a convolution layer will cause the output features to be too smooth and difficult to distinguish. Moreover, the deep neural network is more difficult to train. In this paper, multi-granularity structure and a GCN are combined to represent the node characteristics of the learning network. A multi-granularity structure-based network representation learning method, Multi-GS, is proposed. First, based on the idea of modularity clustering and granular computing, hierarchical multi-granularity space was used to replace the original single-layer network topology space. The GCN model was then used to learn the representation of granules in different coarse- and fine-granularity spaces. Finally, representations of the different grains were combined into representations of nodes in the original space from coarse to fine. Experimental results showed that multi-GS can capture a variety of structural information, including first-order and second-order similarity, intra-community similarity (high-order structure), and inter-community similarity (global structure). In most cases, using multi-granularity structure can improve the classification performance of node classification tasks.

Keywords: network represent learning; network topology; modularity increment; network coarsening; multi-granularity structure; Graph Convolution Network; node classification; link prediction

研究人员常用网络 (图) 描述不同学科领域中实体间的交互关系, 例如生物学领域中的蛋白质互联网络, 社会学领域中的社交网络, 语言学领域中的词共现网络。结合不同的分析任务对网络进行研究、探索, 进而挖掘出隐藏在网络数据中

收稿日期: 2019−05−23.　**网络出版日期:** 2019−09−02.
基金项目: 国家自然科学基金项目 (61876001, 61602003, 61673020); 中国国防科技创新区规划项目 (2017-0001-863015-0009); 国家重点研究与发展项目 (2017YFB1401903); 安徽省自然科学基金项目 (1508085MF113, 1708085QF156).
通信作者: 赵姝. E-mail: zhaoshuzs2002@hotmail.com.

的信息，使之服务于人类。不过，真实场景中的网络数据通常具有稀疏、高维等特质，直接利用这样的数据进行分析通常有较高的计算复杂度，使得许多先进的研究成果无法直接应用到现实的网络环境中。

网络表示学习[1](network representation learning, NRL) 是解决上述问题的有效方法，旨在保留结构信息的前提下，为网络中的每个节点学习一个低维、稠密的向量表示。如此，网络被映射到一个向量空间中，并可通过许多经典的基于向量的机器学习技术处理许多网络分析任务，如节点分类[2]、链接预测[3]、节点聚类[4]、可视化[5]、推荐[6] 等。

网络表示学习不仅要解决网络数据的高维和稀疏的问题，还需使学习到的节点特征表示能够保留丰富的网络结构信息。网络中的节点结构大致分为三类：1) 微观结构，即局部相似性，例如：一阶相似性 (相邻)、二阶相似性 (有共同的邻居)。2) 中观结构，例如：角色相似性 (承担相同的功能)、社团相似性 (属于同一社团)。3) 宏观结构，即全局网络特性，例如：无标度 (度分布符合幂律分布) 特性、小世界 (高聚类系数) 特性。

为获取有效的节点表示，结合最先进的机器学习、深度学习等技术，已提出各种各样的网络表示学习方法。DeepWalk[7] 和 Node2Vec[8] 通过随机游走获取节点的局部相似性。GraRep[9] 和 Walklets[10] 通过将邻接矩阵提升到 k 次幂获取节点的 k 阶相似性。DNGR[11] 通过随机冲浪策略获取节点高阶相似性。Struc2Vec[12] 通过构造层次加权图，并利用层次加权图上的随机游走获取节点的结构相似性。M-NMF[13] 通过融合模块度[14] (modularity) 的非负矩阵分解 (nonnegative matrix factor, NMF) 方法，将社团结构信息纳入网络表示学习中。GraphWave[15] 通过谱图小波的扩散获取节点的角色结构相似性。HARP[16] 通过随机合并网络中相邻节点，迭代地将网络粗化为一系列简化的网络，然后基于这些简化的网络递归地构建节点向量，从而捕获网络的全局特征。

最近，图卷积网络[17](graph convolutional networks, GCN) 越来越受到关注，已经显示 GCN 对网络分析任务性能的改进有着显著的效果。GCN 通过卷积层聚合网络中每个节点及其邻居节点的特征，输出聚合结果的加权均值用于该节点新的特征表示。通过卷积层的不断叠加，节点能够整合 k 阶邻居信息，从而获取更高阶的节点特征表示。尽管 GCN 的设计目标是利用深层模型更好地学习网络中节点的特征表示，但大多数当前方法依然是浅层结构。例如，GCN[18] 实际上只使用两层结构，更多的卷积层甚至可能会损害方法的性能[19]。而且随着模型层数的增加，学习到的节点特征可能过度平滑，使得不同簇的节点变得无法区分。这样的限制违背使用深层模型的目的，导致利用 GCN 模型进行网络表示学习不利于捕获节点的高阶和全局特征。

为克服这种限制，受商空间[20] 中的分层递阶[21] 思想的启发，提出一种基于多粒度结构的网络表示学习方法 (network representation learning based on multi-granularity structure, Multi-GS)。Multi-GS 首先基于模块度[22] 和粒计算[23] 的思想，利用网络自身的层次结构，即社团结构，通过使用局部模块度增量迭代地移动和合并网络中的节点，构造网络的粗粒度结构。利用粗粒度的结构生成更粗粒度的结构，反复多次，最终获得分层递阶的多粒度网络结构。在多粒度结构中，不同粗细的粒能够反映节点在不同粒度空间上的社团内邻近关系。然后，Multi-GS 使用无监督的 GCN 模型分别学习不同粒度空间中粒的特征表示向量，学习到的特征能够反映不同粒度下粒间的邻近关系，不同粗细粒度中的粒间关系能够表示不同阶的节点关系，粒度越粗阶数越高。最后，Multi-GS 将不同粒度空间中学习到的粒特征表示按照由粗到细的顺序进行逐层细化拼接，输出最细粒度空间中拼接后的粒特征表示作为初始网络的节点特征表示。实验结果表明，结合多粒度结构能使 GCN 有效地捕获网络的高阶特征，学习的节点表示可提升诸如节点分类任务的性能。

1　问题定义

定义 1　设网络 $G=(V,E,\boldsymbol{A})$，其中，V 代表节点集合，E 代表边集合，$\boldsymbol{A}$ 代表邻接矩阵。记 $v_i\in V$ 代表一个节点，$e_{ij}=(v_i,v_j)\in E$ 代表一条边，邻接矩阵 $\boldsymbol{A}\in\mathbf{R}^{n\times n}$ 表示网络的拓扑结构，$n=|V|$，若 $e_{ij}\in E$，则 $A_{ij}=w_{ij}>0$；若 $e_{ij}\notin E$，则 $A_{ij}=0$。记 $d(v_i)=\sum A_i$ 代表节点 v_i 的度，$\Gamma(v_i)=\{v_j|e_{ij}\in E\}$ 代表节点 v_i 的邻居节点集合。

定义 2　网络的模块度[22]Q 定义为

$$Q=\frac{1}{2m}\sum_{c\in C}\left(\frac{l_c}{2m}-\left(\frac{D_c}{2m}\right)^2\right) \tag{1}$$

式中：m 表示网络中的边的总数；l_c 表示社团 c 中所有内部边的总数，$\sum_{v_i\in c}A_{ii}$；D_c 表示社团 c 中所有节点的度的总和；C 表示所有社团构成的集合。在社团挖掘任务中，通常使用模块度评价社团划

分的效果, 模块度 Q 值越高, 表明社团划分的效果越佳。

基于式 (1), 可以推导出两个社团合并后的局部模块度增量 ΔQ。设当前划分中的任意两个社团 p 和 q, 合并 p 和 q 后的社团为 k, 产生的局部模块度增量 ΔQ_{pq} 的计算方法如下:

$$\begin{aligned}\Delta Q_{pq} &= Q_k - Q_p - Q_q = \\ &\frac{(l_p+l_q)^2}{2m} - \left(\frac{D_p+D_q}{2m}\right)^2 - \frac{l_p}{2m} + \left(\frac{D_p}{2m}\right)^2 - \\ &\frac{l_q}{2m} + \left(\frac{D_q}{2m}\right)^2 = \frac{l_p}{2m} + \frac{l_{pq}}{m} + \frac{l_q}{2m} - \left(\frac{D_p}{2m}\right)^2 - \\ &\frac{D_pD_q}{2m^2} - \left(\frac{D_q}{2m}\right)^2 - \frac{l_p}{2m} + \left(\frac{D_p}{2m}\right)^2 - \frac{l_q}{2m} + \left(\frac{D_q}{2m}\right)^2 = \\ &\frac{l_{pq}}{m} - \frac{D_pD_q}{2m^2}\end{aligned} \tag{2}$$

其中, $l_{pq} = \sum_{v_i \in p, v_j \in q} A_{ij}$。

定义 3　给定初始网络 $G^{(0)}$。在粒度世界中, 网络中的每个节点可视为一个基本粒。同样用边描述粒间的关系。通过粒度衡量粒的大小 (粗细)。基本粒是指最细粒度的粒。粒化是指将多个粒合并为一个更粗粒度的粒的操作。

将初始网络结构视为由基本粒构成的最细粒度的粒层结构。粒化分层是通过聚合和粒化操作, 迭代地形成粒度由细到粗的多粒层结构。具体地说, 通过不断聚合不同粒层中的粒和边, $G^{(0)}$ 被递归压缩成一系列粒度由细到粗的粒层 $Gr^{(0)}, Gr^{(1)}, \cdots, Gr^{(k)}$, 如图 1 所示。

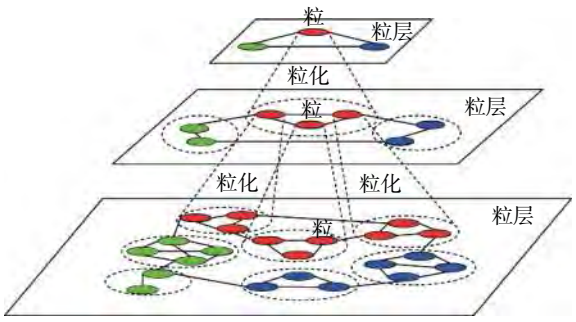


图 1　网络拓扑的粒化分层示例

Fig. 1　An example of hierarchical view of network topology

定义 4　给定网络 G, 网络表示学习的目标是将网络中的节点 $v_i \in V$ 映射到低维向量 $\boldsymbol{z}_i \in \mathrm{R}^d$, 其中, d 表示向量的维度, $d \ll n$。学习到的向量表示可客观反映节点在原始网络中的结构特性。例如, 具有相似结构的节点在特征向量的欧式距离空间中彼此靠近, 不相似的节点彼此远离。

2　网络表示学习方法 Multi-GS

Multi-GS 首先基于模块度构建多粒度的网络分层结构; 接着使用 GCN 模型依次学习不同粒层中所有粒的特征向量; 然后自底向上逐层对粒的特征向量进行映射、拼接; 最后输出最终的结果作为初始网络中节点的特征表示, 如图 2 所示。

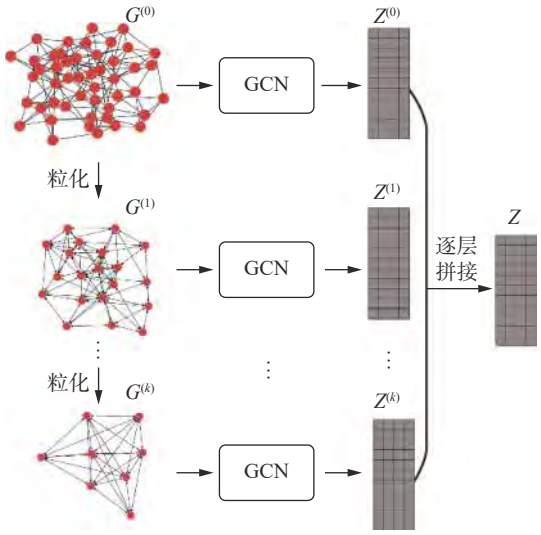


图 2　Multi-GS 方法的框架

Fig. 2　Framework of Multi-GS approach

2.1　基于模块度的网络粒化分层

本小节介绍 Multi-GS 的粒化分层操作。主要包含两个步骤: 1) 粒的移动与合并: 移动和合并的决定取决于局部模块度增量计算结果。2) 粒化: 生成更粗粒度的粒层结构。相关细节见算法 1。

算法 1　网络粒化分层 (Graphgranular)

输入　网络 $G=(V,E)$, 最大粒化层数 k;

输出　由细到粗的粒层 $\mathrm{Gr}^{(0)}, \mathrm{Gr}^{(1)}, \cdots, \mathrm{Gr}^{(k)}$。

1)level = 0;

2)granuleGraph = copy_Graph(G); /*复制图结构*/

3)Q_{new}=modularity(); /*按式 (1) 计算模块度*/

4)repeat

5)$\mathrm{Gr}^{(\mathrm{level})} \leftarrow \mathrm{add_Graph}(\mathrm{granuleGraph})$;

6)$C \leftarrow \{C_i | v_i \in \mathrm{granuleGraph}\}$;

7)repeat

8)$Q_{\mathrm{cur}} = Q_{\mathrm{new}}$;

9)for each v_i in granuleGraph do

10) 将粒 v_i 从自身所在的集合中移出;

11)for v_j in $\Gamma(v_i)$ do

12) 按式 (2) 计算粒 v_i 移入集合 C_j 后的模块度增量 ΔQ;

13)end for

14) 将粒 v_i 并入 max(ΔQ)>0 的粒集合;

15)end for

16)Q_{new}=modularity();

17)untill ($Q_{\mathrm{cur}} = Q_{\mathrm{new}}$)

18)$V^* \leftarrow \{v_i^* | \mathrm{each}\ C_i\ \mathrm{in\ granuleGraph}\}$;

19)$E^* \leftarrow \left\{e_{ij}^* | e_{ij}, v_i \in C_i, v_j \in C_j \text{ and } C_i \neq C_j\right\}$;

20)$W^* \leftarrow \left\{W_{ij}^* | \sum w_{ij}, \text{ if } v_i \in C_i \text{ and } v_j \in C_j\right\}$;

21)granuleGraph ← Graph(V^*, E^*, W^*);

22)level = level +1;

23)untill (level> k)

24)return $\text{Gr}^{(0)}, \text{Gr}^{(1)}, \cdots, \text{Gr}^{(k)}$

算法 1 的主要步骤如下：

粒的移动与合并：首先将当前粒层中的每个粒放入不同的集合中 (第 6 行)。其中，子集合的数量等于当前粒层中粒的数量。遍历所有的粒 (第 9 行)，将当前粒移出自身所在的集合 (第 10 行)，依次移入一个相邻粒的集合中，并依据式 (2) 计算移入后的局部模块度增量 ΔQ(第 11~13 行)。待与所有相邻粒集合的 ΔQ 计算完成后，选择与最大 (正值) 模块度增量相关联的相邻粒集合，并将粒并入该集合中 (第 14 行)。遍历结束后，重新计算模块度 Q (第 16 行)。当未达到模块度的局部极大值时，重复上述步骤 (第 8~16 行)。

在第 2 个步骤中，新粒间的边权由两个对应集合中的粒间的边权和确定。同一集合中粒间的内部边视为新粒的自边。

2.2　基于图卷积网络的粒特征表示学习

本小节介绍 Multi-GS 的 GCN 模型结构，包括两个部分，编码器和解码器，如图 3 所示。GCN 模型借鉴 VGAE[24] 的设计，Multi-GS 不使用节点的辅助信息，仅利用网络的拓扑结构学习节点的特征表示，因此，最终的 GCN 模型在设计上与 VGAE 有所区别。

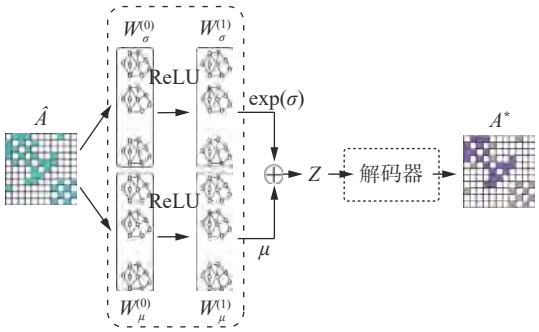


图 3　图卷积神经网络模型结构

Fig. 3　The structure of GCN model

GCN 模型的输入是不同分层中的粒关系矩阵 $\boldsymbol{A}$, $\boldsymbol{A} \in \mathbf{R}^{N \times N}$ 表示同一粒层中粒间的连接关系，其中，N 表示粒的数量。给定粒 i 和粒 j，则 $A_{ij} = w_{ij}$, w_{ij} 表示粒 i 和粒 j 间的连接权重。首先，利用式 (3) 计算得到归一化的矩阵 $\widehat{\boldsymbol{A}} \in \mathbf{R}^{N \times N}$。

$$\widehat{\boldsymbol{A}} = \boldsymbol{D}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \boldsymbol{D}^{-\frac{1}{2}} \tag{3}$$

其中，$\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$, $\boldsymbol{I} \in \mathbf{R}^{N \times N}$ 是单位矩阵，$\boldsymbol{D}$ 是矩阵 $\widetilde{\boldsymbol{A}}$ 的对角矩阵，$D_{ij} = \sum_j \widetilde{\boldsymbol{A}}_{ij}$。GCN 模型的整体结构用式 (4)~(7) 描述：

$$\mu = f\left(\widehat{\boldsymbol{A}} f\left(\widehat{\boldsymbol{A}} \boldsymbol{W}_\mu^{(0)}\right) \boldsymbol{W}_\mu^{(1)}\right) \tag{4}$$

$$\sigma = f\left(\widehat{\boldsymbol{A}} f\left(\widehat{\boldsymbol{A}} \boldsymbol{W}_\sigma^{(0)}\right) \boldsymbol{W}_\sigma^{(1)}\right) \tag{5}$$

$$\boldsymbol{Z} \sim \mathcal{N}(\mu, \exp(\sigma)) \tag{6}$$

$$\boldsymbol{A}^* = \text{sigmoid}\left(\boldsymbol{Z} * \boldsymbol{Z}^{\mathrm{T}}\right) \tag{7}$$

其中，$f(\bullet)$ 表示线性激活函数，第一层使用 RELU 函数，第二层使用 sigmoid 函数；μ 和 σ 分别表示向量矩阵 $\boldsymbol{Z}$ 的均值向量矩阵和标准差向量矩阵；$\boldsymbol{W}_\mu^{(0)}$、$\boldsymbol{W}_\mu^{(1)}$、$\boldsymbol{W}_\sigma^{(0)}$、$\boldsymbol{W}_\sigma^{(1)}$ 是需要训练的权重矩阵；可通过对 μ 和 σ 进行采样得到特征矩阵 $\boldsymbol{Z}$, $\boldsymbol{Z} = \mu + \varepsilon * \exp(\sigma), \varepsilon \sim \mathcal{N}(0, \boldsymbol{I})$；“*”符号表示两个向量的内积。

基于变分推断的编码器，其变分下界的优化目标函数如下：

$$\mathcal{L}_{\text{latent}} = -\left(\frac{0.5}{N}\right) \cdot \text{mean}\left(\sum_{i=1}^{d}\left(1 + 2\log(\sigma) - \mu^2 - \sigma^2\right)\right) \tag{8}$$

解码器使用式 (7) 重构关系矩阵 $\boldsymbol{A}$，对于重构损失，考虑到 $\boldsymbol{A}$ 的稀疏性，使用加权交叉熵损失函数 Loss 构建最终的目标函数，具体公式如下：

$$\begin{aligned}\text{Loss} = \boldsymbol{A} \cdot \left[-\log(\text{sigmoid}(\boldsymbol{A}^*)) * \boldsymbol{W}^{(1)}\right] + \\ (1 - \boldsymbol{A}) \cdot \left[-\log(1 - \text{sigmoid}(\boldsymbol{A}^*))\right]\end{aligned} \tag{9}$$

$$\boldsymbol{W}^{(1)} = \left(N \cdot N - \sum_{i=1}^{N} A_i\right) \Big/ \sum_{i=1}^{N} A_i \tag{10}$$

$$\boldsymbol{W}^{(2)} = N \cdot N \Big/ \left(N \cdot N - \sum_{i=1}^{N} A_i\right) \tag{11}$$

$$\mathcal{L}_{\text{reconst}} = \boldsymbol{W}^{(2)} \text{mean}\left(\text{Loss}\left(\boldsymbol{A}, \boldsymbol{A}^*, \boldsymbol{W}^{(1)}\right)\right) \tag{12}$$

结合式 (8) 和式 (12)，GCN 模型最终的目标函数如下：

$$\mathcal{L} = \mathcal{L}_{\text{latent}} + \mathcal{L}_{\text{reconst}} \tag{13}$$

2.3　算法描述

本小节介绍基于多粒度结构的网络表示学习方法 Multi-GS，主要包括 3 个步骤：利用局部模块度增量 ΔQ，由细到粗地构造多粒度的网络分层结构 (已在 2.1 小节详细介绍)；使用 GCN 模型 (已在 2.2 小节详细介绍) 学习不同粒层中粒的特征表示；将不同粒层的粒特征表示由粗到细地进行逐层拼接，最终得到最细粒度粒层中粒的特征表示；输出此结果作为初始网络的节点特征表示。相关细节见算法 2。

算法 2　基于多粒度结构的网络表示学习 (Multi-GS)

输入　网络 G，粒化层数 k，节点表示向量维度 d，GCN 参数 Θ；

输出　网络表示 $\boldsymbol{Z}$。

1)$Gr^{(0)}, Gr^{(1)}, \cdots, Gr^{(k)} \leftarrow$ Graphgranular(G);
2)for $m = 0$ to k do
3)$\boldsymbol{Z}^{(m)} \leftarrow GCN(Gr^{(m)}.\boldsymbol{A}, \Theta)$;
4)end for
5)for $n = k$ to 1 do
6)$\boldsymbol{Z}^{(n)} \leftarrow$ projection$(\boldsymbol{Z}^{(n)})$;
7)$\boldsymbol{Z}^{(n-1)} \leftarrow$ concatenate$(\boldsymbol{Z}^{(n)}, \boldsymbol{Z}^{(n-1)})$;
8)end for
9)$\boldsymbol{Z} \leftarrow \boldsymbol{Z}^{(0)}$;
10)return $\boldsymbol{Z}$

首先, Multi-GS 算法的输入包括 3 个部分: 网络 G; 粒化层数 k; GCN 模型的参数 Θ, 包括, 节点表示向量维度 d、训练次数和学习率。算法 2 的第 1 行, 使用算法 1 构建多粒度的网络分层结构, 其复杂度为 $O(M+N)$, 其中 M 为每轮迭代中粒的数量, N 是粒层中的边的数量。第 2~第 4 行, 依次将不同粒层的粒关系矩阵 $\boldsymbol{A}$ 作为输入, 利用 GCN 模型学习粒的特征表示。GCN 模型的复杂度与网络的边数呈线性关系, 其复杂度为 $O(mdh)$, 其中 m 是矩阵 $\boldsymbol{A}$ 中非零元素的数量, d 是特征维数, h 是权重矩阵的特征映射数量。另外, 方法还需重建原始拓扑结构, 因此总体复杂度为 $O(mdH+N^2)$, 其中 H 是不同层上所有特征映射的总和。第 5~第 8 行, 将学习到的粒特征向量由粗到细地进行拼接。其中, projection 函数是粒化过程的反向操作。在此过程中, 上层的粒特征向量被映射到一个或多个较细粒度粒特征向量, 投影结束后, 拼接相同粒的两个不同的粒特征表示。循环结束后, 得到基本粒的拼接后的特征表示, 其复杂度为 $O(M)$。第 9 行, 以基本粒的特征表示作为对应节点的特征表示进行输出。

3　实验和结果分析

本节通过节点分类和链接预测任务, 在真实数据集上与 4 个具有代表性的网络表示学习方法进行对比, 验证 Multi-GS 的有效性。实验环境为: Windows10 操作系统, Intel i7-4790 3.6 GHz CPU, 8 GB 内存。通过 Python 语言和 TensorFlow 实现 Multi-GS。

3.1　实验设定

1) 数据集。

实验使用 5 个真实数据集, 包括引文网络、生物网络、词共现网络和社交网络, 详细信息见表 1。Cora[25] 是引文网络。其中, 节点代表论文, 根据论文的不同主题分为 7 类。边代表论文间的引用关系。该网络包含 2 708 个节点和 5 278 条边。Citeseer[25] 同样是引文网络。该网络包含 6 类的 3 312 种出版物。边代表不同出版物间的引用关系, 共有 4 660 条边。PPI[26] 是生物网络。该网络包含 3 890 个节点和 38 739 条边。其中, 节点代表蛋白质, 根据不同的生物状态分为 50 类, 边代表蛋白质间的相互作用。WiKi[27] 是维基百科数据库中单词的共现网络。该网络包含 4 777 个节点, 92 517 条边, 以及 40 种不同的词性标签。BlogCatalog[28] 是来自 BlogCatalog 网站的社交网络。节点代表博主, 并根据博主的个人兴趣划分为 39 类, 边代表博主间的友谊关系。该网络包含 10 312 个节点和 333 983 条边。

2) 对比算法。

实验选择 4 种具有代表性的网络表示学习算法作为对比算法, 包括 DeepWalk、Node2Vec、GraRep、DNGR。关于这些方法的简要描述如下:

DeepWalk[7]: 使用随机游走获取节点序列, 通过 SkipGram 方法学习节点表示。

Node2Vec[8]: 类似于 DeepWalk, 但是使用有偏向的随机游走获取节点序列。

GraRep[9]: 通过构造 k 步概率转移矩阵学习节点表示, 能够保留节点的高阶相似性。

DNGR[11]: 使用随机冲浪方法获取节点的高阶相似性, 利用深度神经网络学习节点表示。

3) 参数设定。

对于 Multi-GS 方法中的 GCN 模型, 使用 Adam 优化器更新训练中的参数, 学习率设为 0.05。对于 DeepWalk 和 Node2Vec, 节点游走次数设为 10, 窗口大小设为 10, 随机游走的长度设为 80。Node2Vec 的参数 p=0.25、q=4。对于 GraRep, k_{step}=4。对于 DNGR 的随机冲浪方法, 迭代次数设为 4, 重启概率 $\alpha = 0.98$, 自编码器的层数设为 2, 使用 RMSProp 优化器, 训练次数设为 400, 学习率设为 0.002。为进行公平比较, 所以方法学习的节点表示维度均设为 128。

表 1　数据集信息
Table 1　Datasets information

数据集	节点数	边数	类别数	平均度
Cora	2 708	5 278	7	3.898
Citeseer	3 312	4 660	6	2.814
PPI	3 890	38 739	50	19.917
WiKi	4 777	92 517	40	38.734
BlogCatalog	10 312	333 983	39	64.776

3.2　节点分类

利用节点分类任务比较 Multi-GS 和对比算

法的性能差异。实验挑选 4 种不同领域数据集，包括 Citeseer、PPI、WiKi 和 BlogCatalog。首先各自使用网络中所有节点学习节点的特征表示，对于 Multi-GS，为比较不同的粒化层次对方法性能的影响，针对不同的数据集，分别设置 5 组实验，在每组实验中，将粒化层次从 0 设到 4(k 为 0~4)。k=0 表示 Multi-GS 不使用多粒度结构进行联合学习表示，仅利用原始网络通过 GCN 模型学习节点的表示。针对节点分类，使用 Logistic 回归分类器，随机从不同数据集中分别选择{10%, 50%, 90%}节点训练分类器，在其余节点上评估分类器的性能。为衡量分类性能，实验采用 Micro-F_1[29] 和 Macro-F_1[29] 作为评价指标。两个指标越大，分类性能越好。所有的分类实验重复 10 次，报告平均结果。表 2~5 分别展示在 Citeseer、PPI、WiKi 和 BlogCatalog 数据集上的节点分类 Micro-F_1 和 Macro-F_1 的均值，其中，粗体表示性能最好的结果，下划线表示对比算法中性能最优的结果。

表 2　Citeseer 数据集上的 Micro-F_1 和 Macro-F_1 结果
Table 2　Micro-F_1 and Macro-F_1 results on Citeseer dataset

对比算法	Micro-F_1			Macro-F_1		
	10%	50%	90%	10%	50%	90%
DeepWalk	0.511	0.591	0.596	0.469	0.540	0.538
Node2Vec	0.531	0.594	0.612	0.480	0.542	0.561
GraRep	0.520	0.546	0.563	0.465	0.486	0.498
DNGR	0.454	0.535	0.545	0.418	0.490	0.492
Multi-GS(k=0)	0.380	0.452	0.489	0.338	0.408	0.433
Multi-GS(k=1)	0.465	0.536	0.573	0.422	0.495	0.534
Multi-GS(k=2)	0.528	0.618	0.658	0.487	0.581	0.627
Multi-GS(k=3)	0.541	0.634	0.666	0.499	0.599	0.632
Multi-GS(k=4)	**0.542**	**0.659**	**0.699**	**0.501**	**0.626**	**0.661**

表 3　PPI 数据集上的 Micro-F_1 和 Macro-F_1 结果
Table 3　Micro-F_1 and Macro-F_1 results on PPI dataset

对比算法	Micro-F_1			Macro-F_1		
	10%	50%	90%	10%	50%	90%
DeepWalk	0.160	0.209	0.232	0.131	0.181	0.190
Node2Vec	0.155	0.204	0.228	0.125	0.177	0.189
GraRep	0.192	0.239	0.254	0.148	0.195	0.201
DNGR	0.145	0.188	0.223	0.126	0.164	0.186
Multi-GS(k=0)	0.173	0.197	0.209	0.128	0.153	0.159
Multi-GS(k=1)	0.181	0.218	0.232	0.135	0.169	0.173
Multi-GS(k=2)	0.188	0.231	0.258	0.143	0.190	0.189
Multi-GS(k=3)	**0.194**	**0.241**	**0.266**	**0.157**	**0.199**	**0.201**
Multi-GS(k=4)	0.177	0.215	0.219	0.139	0.176	0.174

表 4　WiKi 数据集上的 Micro-F_1 和 Macro-F_1 结果
Table 4　Micro-F_1 and Macro-F_1 results on WiKi dataset

对比算法	Micro-F_1			Macro-F_1		
	10%	50%	90%	10%	50%	90%
DeepWalk	0.416	0.473	0.492	0.070	0.089	0.093
Node2Vec	0.422	0.486	0.509	0.078	0.099	0.106
GraRep	0.475	0.523	0.527	0.094	0.117	**0.123**
DNGR	0.341	0.420	0.434	0.066	0.083	0.083
Multi-GS(k=0)	0.472	0.474	0.472	0.078	0.073	0.070
Multi-GS(k=1)	0.469	0.502	0.513	0.071	0.078	0.084
Multi-GS(k=2)	0.476	0.504	0.522	0.081	0.087	0.101
Multi-GS(k=3)	**0.483**	**0.524**	**0.589**	**0.098**	**0.118**	**0.123**
Multi-GS(k=4)	0.416	0.473	0.492	0.070	0.089	0.093

表 5　BlogCatalog 数据集上的 Micro-F_1 和 Macro-F_1 结果
Table 5　Micro-F_1 and Macro-F_1 results on BlogCatalog dataset

对比算法	Micro-F_1			Macro-F_1		
	10%	50%	90%	10%	50%	90%
DeepWalk	0.339	0.397	0.408	0.191	0.253	0.263
Node2Vec	0.341	0.399	0.415	0.200	0.263	0.281
GraRep	0.367	0.400	0.405	0.198	0.235	0.235
DNGR	0.266	0.327	0.342	0.156	0.184	0.190
Multi-GS(k=0)	0.314	0.365	0.363	0.120	0.126	0.120
Multi-GS(k=1)	0.355	0.380	0.394	0.137	0.158	0.173
Multi-GS(k=2)	**0.385**	**0.405**	**0.431**	**0.202**	**0.269**	**0.283**
Multi-GS(k=3)	0.368	0.399	0.417	0.148	0.163	0.179
Multi-GS(k=4)	0.323	0.364	0.380	0.143	0.160	0.172

实验结果显示，在对比算法中，GreRap 表现出强有力的竞争力，Node2Vec 也表现不俗。因无法获取节点的一阶相似性，故 DNGR 表现较差。针对 Multi-GS，可以发现，相对于不使用联合学习 (k=0) 的情形，使用多粒度结构在多数情况下可提升方法的性能，说明保留节点的高阶相似性可提升节点分类任务的性能。对于相同的数据集，Multi-GS 在不同的粒化层次下存在差异。具体来说，在 Citeseer 数据集上，随着粒化层数的增加，Multi-GS 的 Micro-F1 和 Macro-F1 值逐渐增大。在 PPI 和 WiKi 数据集上，最佳的结果出现在 k=3 时。在 BlogCatalog 数据集上，当 k=2 时方法性能最好。依据表 1 中不同数据集平均度的统计结果，可以看出，Citeseer 数据集的平均度是 3.898 1，说明该网络是一个弱关系网络，BlogCata-

log 数据集的平均度高达 64.775 6, 是一个强关系网络。在弱关系网络中, 由于不同社团间的联系较弱, 使得不同粒层中粒的粒度差异较小, 而对于强关系网络, 由于社团内部边的密度与不同社团间的边密度差异较小, 使得小社团快速合并成大社团, 导致不同粒层间的粒度差异会非常大。在强关系网络中, 随着粒化层数的增加, 各粒层中相应粒的特征趋于雷同, 若拼接过多类似的特征将导致节点自身的特征被弱化, 导致 Multi-GS 的性能会有先提升再降低的情况。因此, 针对不同的数据集, 如何设置一个合理的粒化层数是 Multi-GS 需要考虑的问题。

3.3　链接预测

链接预测任务是预测网络中给定节点间是否存在边。通过链接预测可以显示不同网络表示方法的链接预测能力。对于链接预测任务, 仍然选择 Citeseer、PPI、WiKi 和 BlogCatalog 作为验证数据集, 分别从不同数据集中随机移除现有链接的 50%。使用剩余网络, 利用不同的方法学习节点表示。另外, 将被移除边中的节点对作为正样本, 同时随机采样相同数量未连接的节点对作为负样本, 使正样本和负样本构成平衡数据集。实验中, 首先基于给定样本中的节点对, 通过表示向量计算其余弦相似度得分, 然后使用 Logistic 回归分类器进行分类, 并通过曲线下面积[29] (area under curve, AUC) 评估标签间的一致性和样本的相似性得分。对于 Multi-GS, k 为 0~4。表 6 显示链接预测任务中, 不同算法在 Citeseer、PPI、WiKi 和 BlogCatalog 数据集上的 AUC 值, 其中, 粗体表示性能最好的结果, 下划线表示对比算法中性能最优的结果。

表 6　链接预测任务中不同数据集上的 AUC 结果
Table 6　AUC score on all datasets

对比算法	Citeseer	WiKi	PPI	BlogCatalog
DeepWalk	0.635 4	0.757 7		0.724 4
Node2Vec	0.612 1	0.742 0	0.600 1	0.712 6
GraRep			0.616 2	
DNGR	0.651 5	0.756 4	0.516 5	0.717 7
Multi-GS(k=0)	**0.996 5**	**0.925 7**	**0.872 4**	**0.903 6**
Multi-GS(k=1)	0.991 5	0.877 5	0.828 5	0.736 9
Multi-GS(k=2)	0.973 6	0.807 8	0.747 7	0.664 1
Multi-GS(k=3)	0.935 0	0.784 1	0.607 6	0.649 8
Multi-GS(k=4)	0.911 2	0.780 1	0.602 6	0.627 2

表 6 的结果显示, 在对比算法中, GraRep 表现依然最好。对于 Multi-GS, 当不使用联合学习框架时, 方法的性能是最优的。以 AUC 为评价标准, 相对于对比算法中的最优结果, 在 Citeseer 数据集上相对提高 45.24%, 在 WiKi 数据集上相对提高 15.4%, 在 PPI 数据集上相对提高 39.14%, 在 BlogCatalog 数据集上相对提高 20.66%。但是, 随着粒化层数的增加, 对于链接预测任务, 方法的性能会越来越差, 下降速度会随着网络的密度成正比。综合来看, 在链接预测任务中, 利用多粒度结构联合学习到的节点表示无法提升链接预测能力, 说明该类任务需要更多节点自身的特征, 节点低阶相似性比高阶相似性更重要。虽然融合多粒度结构中节点的高阶特征会导致 Multi-GS 性能下降, 但可以看出, 在较低的 k 值下, 仅利用节点的拓扑结构信息, Multi-GS 的链接预测结果十分理想, 说明 Multi-GS 中 GCN 模型能有效地捕获节点的低阶相似性。

3.4　可视化

在本小节中, 对 Multi-GS 和对比算法学习的节点表示利用可视化进行比较。由于空间限制, 实验选择节点数较少的 Cora 作为可视化数据集。其中, 每个节点代表一篇机器学习论文, 所有节点按照论文的主题分为 7 类。实验通过 t-SNE[27] 工具, 将所有方法的节点表示投影到二维空间中, 不同类别的节点用不同的颜色显示。可视化结果如图 4 所示。

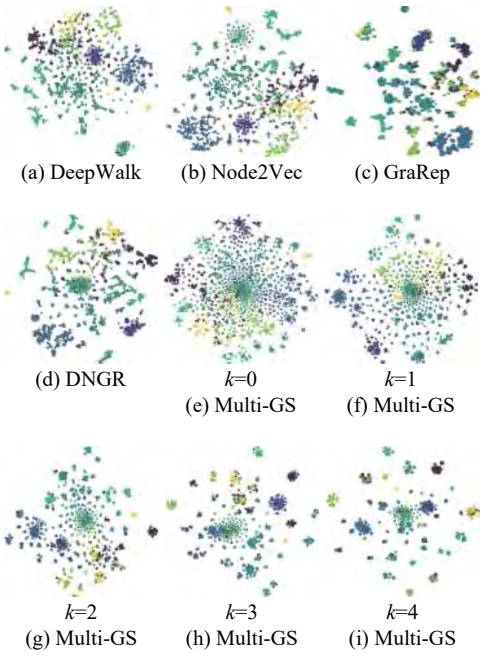


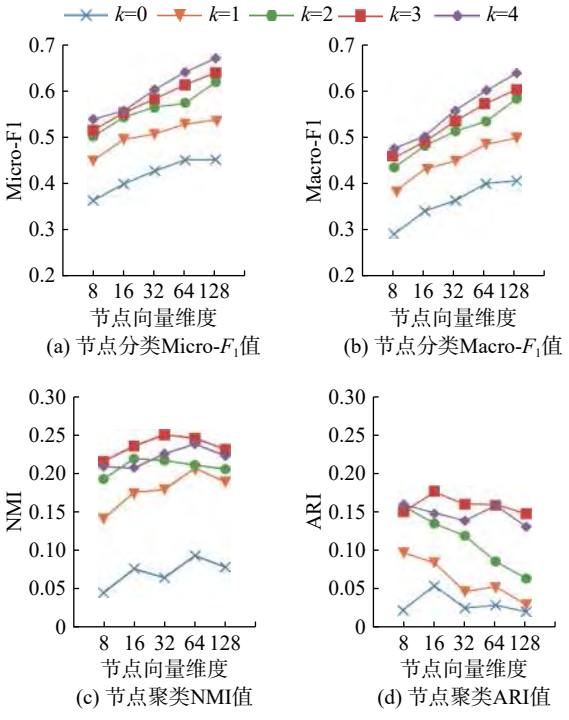
图 4　Cora 数据集的可视化结果
Fig. 4　The visualization of Cora dataset

在图 4 中, DeepWalk、Node2Vec 的表示结果较差, 节点散布在整个空间中, 不同类别的节点相互混在一起, 无法观察到分组结构, 意味着算法无法将相似节点组合在一起。通过 GreRap 的

可视化结果，能够看出节点间的分组结构。对于 Multi-GS，在图 4(e) 中，不同分类的节点相互混合，这种现象在图的中心尤其明显。意味着仅保留低阶相似性的节点表示无法区分不同分类的节点。图 4(f) 显示结果与图 4(e) 相似。在图 4(g)~图 4(i) 中，可以看到节点逐渐开始呈现紧凑的分组结构，而且不同组之间的距离越来越大，随着层数的增加，Multi-GS 可以将相似结构的节点进行分组并推到一起。因此，在节点分类任务中，利用多粒度结构使 Multi-GS 获得更好的结果。

3.5　参数敏感性分析

本节进行参数敏感性实验，主要分析不同的特征维度和粒化层数对 Multi-GS 性能的影响。实验针对 Citeseer 数据集，利用 Multi-GS 在不同粒化层数下学习到的不同维度的节点表示，通过节点分类、节点聚类和链接预测任务对 Multi-GS 进行评估，并报告相关的实验结果。对于节点分类任务，随机选择 50% 节点训练分类器。采用 Micro-F1 和 Macro-F1 作为评价指标。对于节点聚类任务，采用 NMI[30] 和 ARI[30] 作为评价指标。对于链接预测任务，移除 50% 的链接，采用 AUC 作为评价指标。参数 k 表示最终节点的特征表示融合的粒化层数，若 k=0，表示仅选取最细粒度的粒特征表示作为最终的节点特征表示，k=1，表示用第 0 层和第 1 层的粒学习到的特征表示进行拼接后的向量作为最终的节点特征表示，以此类推。其中，0 表示最细粒度，k 值越大，表示粒度越粗。对于所有任务，重复实验 10 次并报告平均结果，实验结果如图 5 所示。



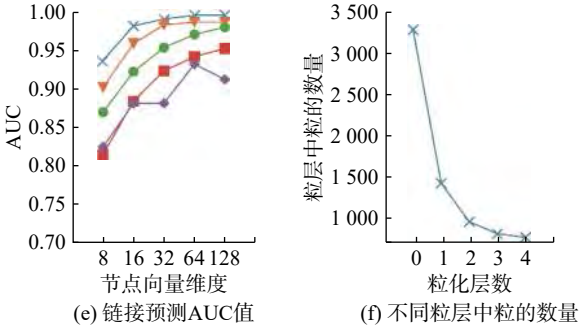


图 5　参数敏感性分析

Fig. 5　Parametric sensitivity analysis

图 5 的结果表明，对于节点分类任务，Micro-F1 和 Macro-F1 指标随着特征维度的上升而上升。因为一个更大的特征维度可以保留网络中更多的信息。随着粒化层数的增加，Micro-F1 和 Macro-F1 指标也逐渐提升，但是可以看出，这样的提升效果会随着粒化层数的增加而越变越小，甚至退化。对于节点聚类任务，NMI 和 ARI 的最优结果都出现在 k=3 时，继续增加层数，结果会下降。对于链接预测任务，AUC 指标随着特征维度的上升而上升，这是合理的情形。但是当 k=4 时，AUC 指标发生波动，说明叠加更多的层次会导致学习的特征表示发生信息变化，这是需要避免的情况。通过图 5(f) 中显示的各层中粒的数量的变化曲线，可以看出，不同层间的粒度差异会随着层数的增加而减少。在第 3 层和第 4 层间，这种粒度差异几乎很小，意味着节点在第 3 层和第 4 层中的特征极为相似，若拼接过多类似的高阶特征向量，导致节点自身的特征被弱化，使得最终 Multi-GS 的输出表示不能在网络分析任务中发挥出方法优势。

4　结束语

在网络表示学习中，如何让学习到的节点特征表示能够保留网络结构的局部和全局特征，仍是一个开放和重要的研究课题。本文结合分层递阶的思想，提出一种无监督网络表示学习方法 Multi-GS，通过构建网络的深度结构解决 GCN 无法有效捕获节点高阶相似性特征的问题。该方法首先利用模块度增量逐步构建网络的多粒度分层结构，然后利用 GCN 模型学习不同粒度空间中粒的特征表示，最后将已学习的粒特征向量逐层映射拼接为原始网络的节点表示。利用 Multi-GS 可捕获多种网络结构信息，包括一阶和二阶相似性、社团内相似性 (高阶结构) 和社团间相似性 (全局结构)。

为验证 Multi-GS 方法的性能, 通过在 4 个真实数据集上进行节点分类任务和链接预测任务, 并与几个经典的网络表示学习方法进行比较。从实验结果上看, 针对节点分类任务, 使用多粒度结构的 Multi-GS 能够改进节点的特征表示, 提升 GCN 模型的节点分类性能。但是由于网络结构的多样性和复杂性, Multi-GS 的粒化层数无法固定, 必须根据不同结构的网络进行调整。针对链接预测任务, 使用多粒度结构 Multi-GS 对 GCN 模型的性能造成损害。说明节点间的低阶邻近关系对链接预测任务是至关重要的。尽管如此, 在不使用多粒度结构的情况下, 以 AUC 为评价指标, 相对于对比算法, Multi-GS 的性能优势非常明显。针对 Multi-GS 超参数敏感性的实验结果可以看出, 面对不同的网络分析任务, 融合不同粒度的粒特征对 Multi-GS 的性能有着不同程度的影响。

未来工作方向包括探索其他网络粒化分层技术和继续深入研究不同的层和不同粗细的粒以及不同类型的网络结构对 Multi-GS 的影响。

参考文献:

[1] 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 Cunchao, YANG Cheng, LIU Zhiyuan, et al. Network representation learning: an overview[J]. Scientia sinica informationis, 2017, 47(8): 980–996.

[2] SHEIKH N, KEFATO Z T, MONTRESOR M. Semi-supervised heterogeneous information network embedding for node classification using 1D-CNN[C]//Proceedings of the Fifth International Conference on Social Networks Analysis, Management and Security. Valencia, Spain, 2018: 177–181.

[3] XU Guangluan, WANG Xiaoke, WANG Yang, et al. Edge-nodes representation neural machine for link prediction[J]. Algorithms, 2019, 12(1): 12.

[4] HU Xuegang, HE Wei, LI Lei, et al. An efficient and fast algorithm for community detection based on node role analysis[J]. International journal of machine learning and cybernetics, 2019, 10(4): 641–654.

[5] PEREDA M, ESTRADA E. Visualization and machine learning analysis of complex networks in hyperspherical space[J]. Pattern recognition, 2019, 86: 320–331.

[6] SHI Chuan, HU Binbin, ZHAO W X, et al. Heterogeneous information network embedding for recommendation[J]. IEEE transactions on knowledge and data engineering, 2019, 31(2): 357–370.

[7] 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, 2014: 701–710.

[8] GROVER A, LESKOVEC J. node2vec: Scalable feature learning for networks[C]//Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, California, USA, 2016: 855–864.

[9] CAO Shaosheng, LU Wei, XU Qiongkai. GraRep: learning graph representations with global structural information[C]// Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. Melbourne, Australia, 2015: 891–900.

[10] PEROZZI B, KULKARNI V, CHEN Haochen, et al. Don't walk, Skip!: Online learning of multi-scale network embeddings[C]//Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017. Sydney, Australia, 2017: 258–265.

[11] CAO Shaosheng, LU Wei, XU Qiongkai. Deep neural networks for learning graph representations[C]//Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 1145–1152.

[12] RIBEIRO L F R, SAVERESE P H P, FIGUEIREDO D R. *struc2vec*: Learning node representations from structural identity[C]//Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Halifax, Canada, 2017: 385–394.

[13] WANG Xiao, CUI Peng, WANG Jing, et al. Community preserving network embedding[C]//Proceedings of the 31th AAAI Conference on Artificial Intelligence. San Francisco, California, USA, 2017: 203–209.

[14] 方莲娣, 张燕平, 陈洁, 等. 基于三支决策的非重叠社团划分 [J]. 智能系统学报, 2017, 12(3): 293–300.
FANG Liandi, ZHANG Yanping, CHEN Jie, et al. Three-way decision based on non-overlapping community division[J]. CAAI transactions on intelligent systems, 2017, 12(3): 293–300.

[15] DONNAT C, ZITNIK M, HALLAC D, et al. Learning structural node embeddings via diffusion wavelets[C]// Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London, UK, 2018: 1320–1329.

[16] CHEN Haochen, PEROZZI B, HU Hifan, et al. HARP: hierarchical representation learning for networks[C]// Proceedings of the 32th AAAI Conference on Artificial Intelligence. New Orleans, Louisiana, USA, 2018: 2127–2134.

[17] ZHANG Si, TONG Hanghang, XU Jiejun, et al. Graph

convolutional networks: algorithms, applications and open challenges[C]//Proceedings of the 7th International Conference on Computational Data and Social Networks. Shanghai, China, 2018: 79–91.

[18] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[C/OL]. [2019-01-28]. https://arxiv.org/pdf/1609.02907.pdf.

[19] LI Qimai, HAN Zhichao, WU Xiaoming. Deeper insights into graph convolutional networks for semi-supervised learning[C]//Proceedings of the 32th AAAI Conference on Artificial Intelligence. New Orleans, Louisiana, USA, 2018: 3538–3545.

[20] 张燕平, 张铃, 吴涛. 不同粒度世界的描述法——商空间法 [J]. 计算机学报, 2004, 27(3): 328–333.
ZHANG Yanping, ZHANG Ling, WU Tao. The representation of different granular worlds: a quotient space[J]. Chinese journal of computers, 2004, 27(3): 328–333.

[21] 赵姝, 柯望, 陈洁, 等. 基于聚类粒化的社团发现算法 [J]. 计算机应用, 2014, 34(10): 2812–2815.
ZHAO Shu, KE Wang, CHEN Jie, et al. Community detection algorithm based on clustering granulation[J]. Journal of computer applications, 2014, 34(10): 2812–2815.

[22] NEWMAN M E J. Fast algorithm for detecting community structure in networks[J]. Physical review E, 2003, 69(6): 066133.

[23] 赵姝, 赵晖, 陈洁, 等. 基于社团结构的多粒度结构洞占据者发现及分析 [J]. 智能系统学报, 2016, 11(3): 343–351.
ZHAO Shu, ZHAO Hui, CHEN Jie, et al. Recognition and analysis of structural hole spanner in multi-granularity based on community structure[J]. CAAI transactions on intelligent systems, 2016, 11(3): 343–351.

[24] KIPF T N, WELLING M. Variational graph auto-encoders[C/OL]. [2019-01-28]. https://arxiv.org/pdf/1611.07308.pdf.

[25] MCCALLUM A K, NIGAM K, RENNIE J, et al. Automating the construction of internet portals with machine learning[J]. Information retrieval, 2000, 3(2): 127–163.

[26] BREITKREUTZ B J, STARK C, REGULY T, et al. The BioGRID interaction database: 2008 update[J]. Nucleic acids research, 2008, 36: D637–D640.

[27] GAO Hongchang, HUANG Heng. Self-paced network embedding[C]//Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. London, UK, 2018: 1406–1415.

[28] TANG Lei, LIU Huan. Relational learning via latent social dimensions[C]//Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Paris, France, 2009: 817–826.

[29] FAWCETT T. An introduction to ROC analysis[J]. Pattern recognition letters, 2006, 27(8): 861–874.

[30] FAHAD A, ALSHATRI N, TARI Z, et al. A survey of clustering algorithms for big data: Taxonomy and empirical analysis[J]. IEEE transactions on emerging topics in computing, 2014, 2(3): 267–279.

作者简介：

张蕾，女，1980 年生，讲师，主要研究方向为数据挖掘、网络表示学习。

钱峰，男，1978 年生，讲师，主要研究方向为数据挖掘、网络表示学习。

赵姝，女，1979 年生，教授，博士生导师，博士，安徽省人工智能学会常务理事，安徽省计算机学会理事，中国人工智能学会粒计算与知识发现专委会委员，CIPS 社会媒体处理专委会委员，主要研究方向为机器学习、粒计算以及社交网络分析和科技大数据挖掘应用研究。获得发明专利和软件著作权多项，发表学术论文 60 余篇。