计算机科学
COMPUTER SCIENCE

语义增强的完全不平衡标签网络表示学习算法

富坤，郭云朋，禚佳明，李佳宁，刘琪

引用本文

富坤，郭云朋，禚佳明，李佳宁，刘琪. 语义增强的完全不平衡标签网络表示学习算法[J]. 计算机科学, 2022, 49(11): 109-116.

FU Kun, GUO Yun-peng, ZHUO Jia-ming, LI Jia-ning, LIU Qi. Semantic Information Enhanced Network Embedding with Completely Imbalanced Labels[J]. Computer Science, 2022, 49(11): 109-116.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210900101

语义增强的完全不平衡标签网络表示学习算法

富　坤　郭云朋　禚佳明　李佳宁　刘　琪
河北工业大学人工智能与数据科学学院　天津 300401
河北省大数据计算重点实验室　天津 300401

摘　要　在网络表示学习的研究中,数据的不完整性问题是一个重要问题,该问题使现有的表示学习算法难以达到预期效果。近年来,不少学者针对此类问题提出了解决方法,这些方法大多仅考虑标签信息本身的缺失问题,对数据不平衡性涉及较少,尤其是某一类别标签完全缺失的完全不平衡问题。解决这类问题的学习算法并不完善,主要存在的问题是在聚合邻域特征时侧重于考虑网络结构信息,未利用属性特征与语义特征间的关系来增强表示结果。为了解决以上问题,提出了融合属性特征与结构特征的 SECT(Semantic Information Enhanced Network Embedding with Completely Imbalanced Labels)方法。首先,在考虑属性空间和语义空间关系的基础上,引入注意力机制进行监督学习,得到语义信息向量;然后,应用变分自编码器无监督提取结构特征以增强算法的鲁棒性;最后,在嵌入空间中融合语义与结构两种信息。将使用 SECT 算法得到的网络向量表示在 Cora,Citeseer 等数据集上进行测试,应用于节点分类任务时与 RECT 和 GCN 等算法相比,取得了 0.86%~1.97%的效果提升。网络向量表示的可视化结果显示,与其他算法相比,SECT 算法的类间距离变大,类簇内部更加紧凑,能较清晰地区分类别边界。实验结果表明了 SECT 算法的有效性,SECT 得益于更好地在低维嵌入空间中融合语义信息,有效提升了存在完全不平衡标签情况下的节点分类任务性能。

关键词:网络表示学习;图嵌入;图注意力网络;完全不平衡标签;变分自编码器
中图法分类号　TP391

Semantic Information Enhanced Network Embedding with Completely Imbalanced Labels

FU Kun,GUO Yun-peng,ZHUO Jia-ming,LI Jia-ning and LIU Qi
College of Artificial Intelligence,Hebei University of Technology,Tianjin 300401,China
Key Laboratory of Big Data Computing,Tianjin 300401,China

Abstract　The problem of data incompleteness has become an intractable problem for network representation learning(NRL) methods,which makes existing NRL algorithms fail to achieve the expected results. Despite numerous efforts have done to solve the issue,most of previous methods mainly focused on the lack of label information,and rarely consider data imbalance phenomenon,especially the completely imbalance problem that a certain class labels are completely missing. Learning algorithms to solve such problems are still explored,for example,some neighborhood feature aggregation process prefers to focus on network structure information,while disregarding relationships between attribute features and semantic features,of which utilization may enhance representation results. To address the above problems,a semantic information enhanced network embedding with completely imbalanced labels(SECT)method that combines attribute features and structural features is proposed in this paper. Firstly,SECT introduces attention mechanism in the supervised learning for obtaining the semantic information vector on precondition of considering the relationship between the attribute space and the semantic space. Secondly,a variational autoencoder is applied to extract structural features under an unsupervised mode to enhance the robustness of the algorithm. Finally,both semantic and structural information are integrated in the embedded space. Compared with two state-of-the-art algorithms,the node classification results on public data sets Cora and Citeseer indicate the network vector obtained by SECT algorithm outperforms others and increases by 0.86%~1.97% under Mirco-F1. As well as the node visualization results exhibit that compared with other algorithms,the vector distances among different-class clusters obtained by SECT are larger,the clusters of same class are more compact,and the class boundaries are more obvious. All these experimental results demonstrate the effectiveness of SECT,which mainly benefited from a better fusion of semantic information in the low-dimensional embedding space,thus extremely improves the performance of node classification tasks under completely imbalanced labels.

Keywords　Network representation learning,Graph embedding,Graph attention network,Completely imbalanced label,Variational autoencoders

---

到稿日期:2021-09-13　返修日期:2022-02-26
基金项目:国家自然科学基金(62072154)
This work was supported by the National Natural Science Foundation of China(62072154).
通信作者:富坤(fukun@hebut.edu.cn)

1 引言

网络表示学习是为网络中每个节点学习一个低维、稠密的向量表示，同时保留较多的网络信息[1-2]。随着信息技术的普及和数据量日益激增，网络数据正扮演着越来越重要的角色。网络数据的增多产生了如下问题：首先，由于法律经济等方面的问题，无法保证所获取数据的完整性；其次，数据集标注较为依赖人力标注，产生的成本与误差不容忽视。这些问题对网络表示学习提出了新的挑战，因此研究标签缺失和标签类别缺失问题具有较大的现实意义。本文主要研究在数据集标签完全不平衡(Completely Imbalanced)情况下的表示学习，针对在学习过程中难以将节点属性特征表征为有效辅助监督信息的问题，进行算法优化并增强算法的鲁棒性。下面简述目前国内外的研究状况。

早期的表示学习算法使用网络数据的谱特征构建关系矩阵，通过分解关系矩阵对高维流形数据进行降维，如 Isomap (Isometric Feature Mapping)[3]，LLE(Locally Linear Embedding)[4]，LE(LaPlacian Eigenmaps)[5]等，其由于复杂度较高难以应用于大规模网络。随着深度神经网络的发展，出现了基于随机游走的 DeepWalk[6]和 Node2Vec[7]、应用于大规模网络上考虑了一阶和二阶相似性的 LINE[8]算法以及基于矩阵分解的 GraRep[9]。这些算法的特点是：使用在网络内路径序列上的中心节点与上下文间的共现频率来学习节点表示向量，侧重于从网络拓扑的角度来提取网络信息。在真实数据集(如引文网络等)中，网络节点包含着丰富的属性信息，而在增强学习效果时这些特征却并未得到充分的利用。因此，利用网络中标记数据和节点属性的算法，可取得最高的性能。该类算法主要有图卷积神经网络 GNN(Graph Neural Network)类表示学习方法，如 GCN(Graph Convolutional Network)[10]，GraphSage[11]，GAT (Graph Attention Networks)[12]，APPNP[13]等。此类算法通常利用网络中已标记的部分标签做半监督学习，并把属性信息基于拓扑关系做邻域聚合。以上算法均是基于数据集中平衡标签的条件，而当数据集中某一类别出现缺失情况时，模型便无法提取缺失类别节点的属性特征。

同一类别的节点属性具有相近性，可以取属性的均值作为类的语义特征。受半监督算法的启发，RECT[14]通过引入这种语义特征提供了额外的监督条件，并取得了较好的结果。但仍存在如下问题：首先，仅使用属性均值表征语义信息并不完全合理，应该引入分布来更好地表示语义信息；然后，语义信息学习的聚合过程仅从结构角度确定邻域聚合权重并不合适；最后，可选用更优的算法进行无监督网络结构信息的学习。

本文针对以上 3 个问题，提出了融合属性特征与结构特征的 SECT 方法，该模型在保留了 RECT 优势的基础上进行了改进。首先，在预处理过程中利用节点的属性特征表征语义信息并考虑高斯先验分布，以采样后的语义向量进行监督学习。然后，在学习过程中，在第一个学习通道用注意力网络学习语义特征；在第二个学习通道使用泛化误差更小的变分自编码器无监督学习网络全局结构特征。最后，将两个通道所学习的向量拼接起来得到最终的节点向量表示，该向量表示结果在低维嵌入空间可保留丰富的语义与结构特征。

2 相关工作

DeepWalk 首次应用了深度学习算法来学习节点低维表示。该算法通过随机游走，采集反映网络拓扑结构的序列信息，对采集到的点序列用 Skip-gram[15]和 Hierarchical Softmax[16-17]求节点的共现概率，得到序列内节点的极大似然估计。通过随机梯度下降更新参数，得到节点的低维实值表示。

Yang 等[18]证明了 DeepWalk 实际上是分解一个 $\boldsymbol{M}$ 矩阵，其中每个元素如式(1)所示：

$$\boldsymbol{M}_{ij}=\frac{\log\left[\boldsymbol{e}_i(\boldsymbol{A}+\boldsymbol{A}^2+\cdots+\boldsymbol{A}^t)\right]_j}{t} \tag{1}$$

其中，$\boldsymbol{A}$ 是转移矩阵；$\boldsymbol{e}_i$ 是指示向量，只在第 i 个位置为 1，其余位置全为 0；t 为随机游走的步数；$\boldsymbol{M}_{ij}$ 表示 v_i 经 t 步游走到 v_j 的概率的对数值，反映节点间的 PMI 值。

GraRep 算法通过矩阵分解来学习网络节点表示，通过邻接矩阵和度矩阵定义一个概率转移矩阵，可以将其看作是一个收缩的邻接矩阵。先对矩阵进行 SVD 分解得到 k 步网络表示，再将 k 步网络表示拼接起来，得到表示能力更强的节点表示。

上述方法从结构角度建模，并未使用节点的属性信息来提升表示性能，因此这类方法的学习性能通常低于利用了属性信息的算法。

VGAE(Variational Graph Auto-encoders)[19]变分自编码器是一种有效的无监督学习算法，该算法在图数据上应用了变分自编码器。算法的结构包含一个编码器和一个解码器，输入节点属性矩阵和邻接矩阵，编码器可学习节点低维向量表示的分布，由学习到的分布生成采样信息，解码器将原样本复原。

图卷积网络(GCN)的卷积操作定义了一个卷积核函数。将节点属性信息通过傅里叶变换到谱频域，在谱频域内完成卷积操作后再反变换回原始域。对离散信息可使用拉普拉斯矩阵的特征值和特征向量组成矩阵以进行离散傅里叶变换。多层 GCN 网络在第一层输入节点属性矩阵。

图注意力网络(GAT)可看作是对 GCN 的改进。GCN 进行卷积操作时中心节点不断聚合其邻域节点的特征，聚合的权重与拓扑结构有关，较为固定。而 GAT 的不同之处在于当聚合邻域特征时为其邻域节点学习不同的聚合权重。上述模型在聚合邻域信息后得到节点的向量表示，由于在嵌入空间中融合了属性特征与拓扑特征，得到的表示结果比仅在嵌入空间融合拓扑特征的算法更好。

WANG 等提出的 RECT 模型解决了标签类别缺失的问题，并定义这类问题为标签信息完全不平衡问题。RECT 模型应用了 GCN 方法去学习网络的结构信息及标签类别的语义信息，并取得了相对不错的效果。然而，对结构信息的学习还有提升的空间。此外，在获取标签类别语义特征时没有考虑属性信息空间与语义信息空间的内在关联性。

3 SECT 模型

为了更好地将语义特征作为半监督信息，模型使用注意

力层学习语义信息；为了实现更小的泛化误差，使用变分自编码器学习网络结构特征。依据以上改进设计了 SECT 框架，模型概图如图 1 所示。该模型由两部分组成：一个结构信息学习模块，生成保留结构信息的向量$\boldsymbol{Z}_1$；另一部分是标签语义信息模块，学习标签语义信息的向量$\boldsymbol{Z}_2$。输入的图 G 是不完整信息图，网络内的节点仅含一定比例的标签且缺失若干标签类别。采用与 Deepwalk 算法等价的 $\boldsymbol{M}$ 矩阵代替邻接矩阵，可使不同类别向量在嵌入空间中具有较大的类间距离[14]。聚合模块对两个通道上学习到的两个向量$\boldsymbol{Z}_1$ 与$\boldsymbol{Z}_2$ 进行聚合，在实验中采用拼接聚合的方式。

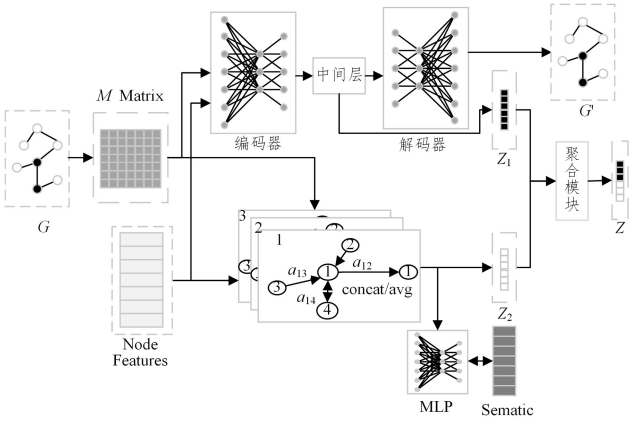

图 1　SECT 算法框架图

Fig. 1　SECT algorithm diagram

3.1　结构信息学习模块

第一个通道是结构信息学习模块，学习保留结构信息的低维表示向量。这一模块把复原的样本与原样本进行比较并考虑分布的影响，隐层的向量便是输入样本的低维表示。

变分自编码器的原理如图 2 所示。输入为邻接矩阵 $\boldsymbol{A}$ 和属性 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$。编码器用两层 GCN 网络实现。

第一层网络如式(2)所示：

$$\boldsymbol{H}=\mathrm{ReLU}(\boldsymbol{AXW}_0) \tag{2}$$

其中，$\boldsymbol{A}$ 是拉普拉斯矩阵，该层学到每个样本对应的低维向量表示 $\boldsymbol{H}=\{\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_n\}$。

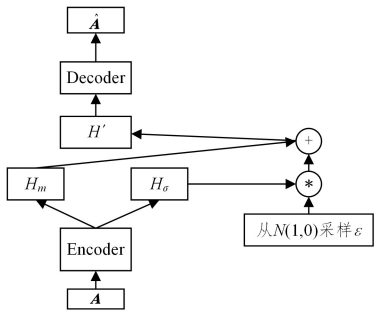

图 2　结构信息模块概图

Fig. 2　Overview of structural information module

第二层网络如式(3)所示：

$$\begin{cases}\boldsymbol{H}_\mu=\boldsymbol{A}\mathrm{ReLU}(\boldsymbol{AXW}_0)\boldsymbol{W}_1^\mu\\ \boldsymbol{H}_\sigma=\boldsymbol{A}\mathrm{ReLU}(\boldsymbol{AXW}_0)\boldsymbol{W}_1^\sigma\end{cases} \tag{3}$$

式(3)用于分别计算 $\boldsymbol{H}$ 的均值μ 和方差σ。

由于采样操作不能直接用梯度下降更新，使用了重参数技巧[20]来避免这一问题。在分布 $N(\mu,\sigma)$上采样$\boldsymbol{H}'$，等价于在分布 $N(0,1)$中采样一个 ε，得$\boldsymbol{H}'=\mu+\varepsilon*\sigma$，则有$\boldsymbol{H}'=\{\boldsymbol{h}_1',\boldsymbol{h}_2',\cdots,\boldsymbol{h}_n'\}$。将向量 $\boldsymbol{H}'$输入解码器 Decoder，复原的邻接矩阵为 $\hat{\boldsymbol{A}}=\mathrm{sigmoid}(\boldsymbol{H}'\boldsymbol{H}'^T)$。

模型中的损失函数如下：首先计算重构损失，其计算过程如式(4)所示，$\boldsymbol{A}$ 是原邻接矩阵，$\hat{\boldsymbol{A}}$ 是由解码器生成的复原邻接矩阵；接着计算 KL 散度，其计算过程如式(5)所示，该式保证隐层表示学习到的分布是一个正态分布；最终的损失函数由 L_D 和 L_{μ,σ^2} 构成，其计算式如式(6)所示，最小化该损失函数 L_{vae}学习得到合适的$\boldsymbol{H}'$向量。

$$L_D=loss(\hat{\boldsymbol{A}},\boldsymbol{A})=loss_1(\boldsymbol{H}'\boldsymbol{H}'^{\mathrm{T}},\boldsymbol{A}) \tag{4}$$

$$L_{\mu,\sigma^2}=\frac{1}{2}\sum_{i=1}^{K}(\mu_i^2+\sigma_i^2-\log\sigma_i^2-1) \tag{5}$$

$$L_{\mathrm{vae}}=L_D-L_{\mu,\sigma^2} \tag{6}$$

本文设计的结构信息学习模块的损失函数如式(6)所示并考虑了图的 $\boldsymbol{M}$ 矩阵重构损失，其计算式如式(7)所示：

$$L_{\mathrm{structure}}=loss_1(\boldsymbol{H}'\boldsymbol{H}'^{\mathrm{T}},\boldsymbol{M})-L_{\mu,\sigma^2} \tag{7}$$

变分自编码器训练过程中通过更新参数$\boldsymbol{W}_0$，$\boldsymbol{W}_1^\mu$ 和$\boldsymbol{W}_1^\sigma$ 来最小化损失函数，其中$\boldsymbol{W}_0$ 是共享参数。图变分自编码器的效果好于传统 GCN 的原因可从模型泛化性与稳定性的角度进行理论验证，由 Bousquet 等[21]提出的定理给出了泛化上界。在此之前先定义替换样本假设稳定性(Hypothesis Stability)[22]，如定义 1 所示。

定义 1　若学习算法 $\mathfrak{L}$ 满足：

$$\boldsymbol{E}_{\boldsymbol{D},\boldsymbol{z}'_{i\mid i^{m+1}}}(|\ell(\mathfrak{L}_{\boldsymbol{D}},\boldsymbol{z}_i)-\ell(\mathfrak{L}_{\boldsymbol{D}^{i,z_i'}},\boldsymbol{z}_i)|)\leqslant\beta \tag{8}$$

其中，$i\in[0,m]$，则称在数据集 D 上该算法 $\mathfrak{L}_D$ 具有关于损失函数 ℓ 的 β 替换样本假设稳定性，则关于泛化上界的定理如定理 1 所示。

定理 1　给定学习算法 $\mathfrak{L}$ 和数据集 $D=\{z_1,z_2,\cdots,z_m\}$，假设损失函数为 $\ell(.,.)\in[0,M]$，若学习算法 $\mathfrak{L}$ 具有替换样本 β-均匀稳定性，则 $\delta\in(0,1)$，此时的泛化以至少 $1-\delta$ 的概率有：

$$R(\mathfrak{L}_D)\leqslant\hat{R}(\mathfrak{L}_D)+\beta+(2m\beta+M)\sqrt{\frac{\ln\left(\frac{1}{\delta}\right)}{2m}} \tag{9}$$

其中，$\hat{R}(\mathfrak{L}_D)$是经验风险，$R(\mathfrak{L}_D)$是泛化风险。

标签类别缺失问题可视作对数据集作样本替换，即将部分含标签的节点替换成不含标签的节点。由定理 1 可知，损失函数为 $\ell(.,.)\in[0,M]$，M 越大，泛化风险的上界便越大。下面确定损失函数的 M 值，应用复原矩阵与原矩阵的 MSELoss[14]计算损失，其上界为预测全错的计算结果。损失函数依据式(4)－式(6)计算，设有 n 个节点，L_D 的上界为 $M_D=\sup\{L_D\}=n^2$，L_{vae}的上界为 $M_{\mathrm{vae}}=\sup\{L_{\mathrm{vae}}\}=n^2-a$，$a$ 为此时 L_{μ,σ^2} 的值。由于 $a\geqslant0$，则可知 $M_{\mathrm{vae}}\leqslant M_D$。依据以上分析，$M$ 的上限即为 M_D 的上限 n^2。通过分析实验中的经验误差可知，VGAE 的泛化上界小于 GCN 模型，因此在分类任务中效果更好。

3.2　标签语义信息学习模块

由于部分属性信息存在于可见类别与不可见类别的交集

中，如能在向量表示中更好地保留语义信息，便能在后续任务中取得更好的效果。

第二个通道是学习标签语义信息向量的模块。使用图注意力网络 GAT 来更好地将标签的语义信息融合到节点嵌入空间中。图注意力网络是将注意力机制应用于图数据中，是对 GCN 的改进。GCN 的卷积操作可看作节点信息的傅里叶变换过程，如式(10)所示：

$$\boldsymbol{Z}=\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{\Theta} \tag{10}$$

其中，$\boldsymbol{A}$ 是拉普拉斯矩阵 $\boldsymbol{A}=\boldsymbol{A}+\boldsymbol{I}_N$，$\boldsymbol{D}^{-\frac{1}{2}}$ 是对角阵，对角线元素是 $\boldsymbol{A}$ 的每行之和，$\boldsymbol{X}$ 是节点属性矩阵，$\boldsymbol{\Theta}$ 是待训练的参数矩阵。

在聚合邻居节点过程中，GCN 为中心节点的每个邻居节点分配相同的权重。实际上，每个邻居节点对中心节点表示的贡献是不同的，因此对不同邻居分配不同的权重较为合理，这点在相关实验中也得到了验证。GAT 的不同点在于给每个节点的邻居节点计算一个注意力值，体现了邻居节点对当前节点表示的贡献度。通过迭代让每个节点不断融合其邻居节点的表示向量，与 GCN 算法得到的结果相比，最后输出的向量表示更好地考虑了邻居节点的影响。

网络图数据是一种非欧几里得结构，需定义一个作用于中心节点邻居上的卷积核函数，这里将式(10)中的卷积核函数 $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ 作用于邻居集合的属性集。如给定图 Graph，假设其有 n 个节点，它的 $n*n$ 的卷积核为：

$$\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}=\begin{bmatrix}\frac{a_{11}}{d_1} & \frac{a_{12}}{d_2} & \cdots & \frac{a_{1n}}{d_n}\\ \frac{a_{21}}{d_1} & \frac{a_{22}}{d_2} & \cdots & \frac{a_{2n}}{d_n}\\ \vdots & \vdots & \vdots & \vdots\\ \frac{a_{n1}}{d_1} & \frac{a_{n1}}{d_2} & \cdots & \frac{a_{n1}}{d_n}\end{bmatrix} \tag{11}$$

其中，a_{ij} 取 0 表示无边，1 表示有边，d_i 是第 i 个节点的度，$i,j\in[1,n]$。

假设每个节点属性表示为 m 维向量，该图 n 个节点的属性叠成 $n*m$ 维的属性矩阵 $\boldsymbol{X}$。

$$\boldsymbol{X}=\begin{bmatrix}x_{11} & x_{12} & \cdots & x_{1m}\\ \vdots & \vdots & \vdots & \vdots\\ x_{n1} & x_{n2} & \cdots & x_{nm}\end{bmatrix}=\begin{bmatrix}\boldsymbol{X}_1\\ \boldsymbol{X}_2\\ \vdots\\ \boldsymbol{X}_n\end{bmatrix} \tag{12}$$

其中，$\boldsymbol{X}_i=[x_{i1},x_{i2},\cdots,x_{im}]$。

那么：

$$\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{X}=\begin{bmatrix}\frac{a_{11}}{d_1}*\boldsymbol{X}_1+\frac{a_{12}}{d_2}*\boldsymbol{X}_2+\cdots+\frac{a_{1n}}{d_n}*\boldsymbol{X}_n\\ \vdots\\ \frac{a_{n1}}{d_1}*\boldsymbol{X}_1+\frac{a_{n2}}{d_2}*\boldsymbol{X}_2+\cdots+\frac{a_{m}}{d_n}*\boldsymbol{X}_n\end{bmatrix} \tag{13}$$

任取一行如式(14)所示：

$$\frac{a_{i1}}{d_1}*\boldsymbol{X}_1+\frac{a_{i2}}{d_2}*\boldsymbol{X}_2+\cdots+\frac{a_{in}}{d_n}*\boldsymbol{X}_n \tag{14}$$

它表示对第 i 个节点，将周围邻居节点的属性聚合到了 i 节点，即以周围邻居的属性乘系数再求和后表示 i 节点的属性。

将式(13)输入全连接网络，即乘 $\boldsymbol{W}$ 矩阵，$\boldsymbol{W}$ 是全连接网络的权重，为待训练参数。若 label 共有 k 类，则 $\boldsymbol{W}$ 是 $m*k$ 维的，那么 $\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{W}$ 便是一个 $n*k$ 维的矩阵。通过分析表明，GCN 实质上是聚合邻居节点的过程。聚合过程中，权重是度的倒数，为一个常数，这种权重难以很好地反映节点间的联系。

GAT 的注意机制是在聚合特征信息时，将注意机制用于确定节点邻域的权重。GAT 的图卷积运算定义为：

$$\boldsymbol{h}_i^t=\sigma\Big(\sum_{j\in N_i}a(\boldsymbol{h}_i^{t-1},\boldsymbol{h}_j^{t-1})\boldsymbol{W}^{t-1}\boldsymbol{h}_j^{t-1}\Big) \tag{15}$$

其中，$\boldsymbol{h}_i^{t-1}$ 是第 $t-1$ 层的 i 节点向量；$\boldsymbol{W}^{t-1}$ 是第 $t-1$ 层网络的权重矩阵；$a()$ 是一个注意力函数，它计算相邻节点 j 对节点 i 的注意力值；σ 是 log softmax 激活函数。

$$e_{ij}=a(\boldsymbol{W}\boldsymbol{h}_i,\boldsymbol{W}\boldsymbol{h}_j) \tag{16}$$

$$\alpha_{ij}=\mathrm{softmax}_j(e_{ij})=\frac{\exp(e_{ij})}{\sum_{k\in N_i}\exp(e_{ik})} \tag{17}$$

$$\alpha_{ij}=\frac{\exp(\mathrm{LeakyReLU}(a^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{h}_i\,\|\,\boldsymbol{W}\boldsymbol{h}_j]))}{\sum_{k\in N_i}\exp(\mathrm{LeakyReLU}(a^{\mathrm{T}}[\boldsymbol{W}\boldsymbol{h}_i\,\|\,\boldsymbol{W}\boldsymbol{h}_j]))} \tag{18}$$

由式(14)可知，聚合邻居节点的过程中，依据重要程度分配不同的权重。在加入注意力机制后，聚合过程变为：

$$\boldsymbol{h}_i=\alpha_{i1}*\boldsymbol{h}_1+\alpha_{i2}*\boldsymbol{h}_2+\cdots+\alpha_{il}*\boldsymbol{h}_l \tag{19}$$

由式(16)可知，权重与节点属性是直接相关的，两节点的属性越接近，对应边的权重越大，聚合后的向量表示就越接近。

为了学习不同子空间中的注意力权重，GAT 还可以使用多头(Multi-head)注意力机制。

$$\boldsymbol{h}_i^t=\Big\|_{k=1}^{K}\sigma\Big(\sum_{j\in N_i}a_k(\boldsymbol{h}_i^{t-1},\boldsymbol{h}_j^{t-1})\boldsymbol{W}_k^{t-1}\boldsymbol{h}_j^{t-1}\Big) \tag{20}$$

其中，K 是注意力头数，a_k 是两节点间的第 k 个注意力值，$\|$ 是拼接操作。

定义 2 对于每个可见的类标签 c，使用类节点属性分布采样作为语义信息向量。

$$\boldsymbol{\mu}_c^s=\mathrm{mean}(\{\boldsymbol{x}_i\mid\forall i,\mathscr{C}_i^s=c\}) \tag{21}$$

$$\boldsymbol{\sigma}_c^s=\sqrt{\mathrm{mean}\{(\boldsymbol{x}_i-\boldsymbol{\mu}_c^s)^2\mid\forall i,\mathscr{C}_i^s=c\}} \tag{22}$$

在标准高斯分布 $N(0,1)$ 中采样 ε。

$$\hat{\boldsymbol{y}}_c=\boldsymbol{\mu}_c^s+\varepsilon*\boldsymbol{\sigma}_c^s \tag{23}$$

其中，$\mathscr{C}_i^s$ 是第 i 个节点的可见(S)类别标签，$\boldsymbol{x}_i$ 是 i 节点的属性向量，$\boldsymbol{\mu}_c^s$ 和 $\boldsymbol{\sigma}_c^s$ 分别为属性的均值与方差，$\hat{\boldsymbol{y}}_c$ 是语义信息。

节点的属性特征是分类的重要依据，以引文网络为例，节点的属性特征是文章的内容，是文章间相互区分的依据。语义信息表征同类别中各节点所含的共性信息，学习这类信息有助于改善标签缺失问题的分类效果。

语义信息的学习实际上是建立从节点向量空间到语义向量空间的映射过程，$F:\boldsymbol{Z}^q\rightarrow\boldsymbol{S}^p$，$\boldsymbol{Z}^q$ 为节点向量，S^p 为语义向量。

使用 MLP 网络在节点向量空间与语义空间建立线性关系，计算式如下：

$$\begin{aligned}&\{\hat{\boldsymbol{y}}'_{\mathscr{C}_i^s}=\boldsymbol{W}_{zs}\boldsymbol{h}_i+b_{zs}\\ &\hat{\boldsymbol{y}}'_{\mathscr{C}_i^s}=\boldsymbol{W}_{zs}(\alpha_{i1}\boldsymbol{h}_1+\alpha_{i2}\boldsymbol{h}_2+\cdots+\alpha_{il}\boldsymbol{h}_l)+b_{zs}\end{aligned} \tag{24}$$

其中，$\hat{\boldsymbol{y}}'_{\mathscr{C}_i^s}$ 是预测的标签语义信息向量，语义信息学习模块的

损失函数如式(22)所示：

$$L_{\text{semantic}}=\sum_{i\in K} loss(\hat{\boldsymbol{y}}'_{c_i^s},\hat{\boldsymbol{y}}'_{c_i^s}) \tag{25}$$

其中，K 为训练集节点个数，$\hat{\boldsymbol{y}}'_{c_i^s}$ 是真实的标签语义信息向量。

训练过程中更新了注意力网络的 $\boldsymbol{W}_{\text{attention}}$ 和 c 值，MLP 网络的 $\boldsymbol{W}_{bs}$ 和 b_{bs}，并最小化了损失 L_{semantic}。

3.3 SECT 算法

SECT 算法如算法 1 所示，其中各符号的含义如下：$\boldsymbol{A}$ 表示邻接矩阵，$\boldsymbol{A}$ 表示拉普拉斯矩阵，$\boldsymbol{x}$ 表示节点特征向量，$\boldsymbol{Z}$ 表示低维向量，c 表示可见类别标签，v 表示图中的节点，$\boldsymbol{M}$ 表示 deepwalk 的等价矩阵，$\boldsymbol{h}$ 表示低维向量，K 表示节点个数。

算法 1　SECT 算法

输入：G(v,e)，**x**，**A**，c

输出：节点向量表示 **Z**//结构信息学习模块

1. 根据式(1)计算 **M**←**A** //计算 **M** 矩阵
2. for i=1 to K
3. $\mathbf{h}_i$←ReLU(**A** $\mathbf{x}_i$ $\mathbf{W}_0$) //计算节点低维表示$\mathbf{h}_i$
4. end
5. $\mathbf{h}_\mu$←**A**{$\mathbf{h}_i$,i∈(0,K)}$\mathbf{W}_1^\mu$//计算节点低维表示均值
6. $\mathbf{h}_\sigma$←**A**{$\mathbf{h}_i$,i∈(0,K)}$\mathbf{W}_1^\sigma$//计算节点低维表示方差
7. 利用式(4)计算$\mathbf{h}_i$//依据学习的节点低维表示分布采样
8. 计算 $\hat{\mathbf{M}}$←$\mathbf{H}'\mathbf{H}'^{\mathrm{T}}$//复原 **M** 矩阵
9. min($L_{\text{structure}}$)得到$\mathbf{Z}_1$，更新$\mathbf{W}_0$，$\mathbf{W}_1^\mu$，$\mathbf{W}_1^\sigma$//语义信息学习模块
10. 由定义 2 计算$\hat{\mathbf{y}}_c^s$←$\boldsymbol{\mu}_c^s+\varepsilon*\boldsymbol{\sigma}_c^s$//计算语义信息
11. for i=1 to K
12. for j∈N(v_i)//N(v_i)是 i 节点的邻居集合
13. α_{ij}←a($\mathbf{W}\mathbf{x}_i$,$\mathbf{W}\mathbf{x}_j$) //计算注意力权重
14. end
15. 依据式(19)$\mathbf{h}_i$←aggregate({$\mathbf{h}_u$,∀u∈N(v_i)})//聚合邻居节点信息
16. 由式(24)计算$\hat{\mathbf{y}}'_c$//计算语义信息的估计
17. end
18. min(L_{semantic})得到$\mathbf{Z}_2$，更新 $\mathbf{W}_{\text{attention}}$，a，$\mathbf{W}_{zs}$，$\mathbf{b}_{zs}$
19. **Z**←concatenate($\mathbf{Z}_1$,$\mathbf{Z}_2$) //拼接$\mathbf{Z}_1$，$\mathbf{Z}_2$得到最终节点表示 **Z**

4　实验设计

网络分析中最常见的任务是节点分类任务。举例来说，在社交网络中往往按兴趣标签对用户进行分类，而在蛋白质网络中根据蛋白质的功效和成分进行划分；引文网络中，对论文的类别进行划分。由于网络规模普遍较大，标注信息往往是稀疏的，因此需要设计算法，利用拓扑结构信息、属性信息以及少量的已标注节点信息，来对大量未标注节点的分类情况进行标注。实验使用 SECT 模型的网络表示学习结果进行节点分类任务，计算了分类或多分类任务的 Mirco-F1 值以及单标签分类的准确率，并进行了降维可视化实验。

4.1 数据集及评价方法

本文使用了 3 个数据集，分别是表 1 中所列的 Cora，Citeseer 及 PPI 数据集。Cora 与 Citeseer 是引文网络数据集，Cora 中将 2708 篇科学文献分成 7 个类，Citeseer 将 3312 篇文献分为 6 个类，网络中的节点是科学文献，边是文献间的引用关系。此外，我们还在 PPI 数据集上进行实验，PPI 是生物图数据集。PPI 数据集中的节点具有多个标签，同时节点没有属性信息。由于图神经网络尤其图卷积模型中需要属性信息，为了更好地适应分类的测试任务，在实验中将邻接矩阵的行作为节点的属性信息[14]。

表 1　3 个数据集的简介

Table 1　Illustration of 3 datasets

数据集	Citeseer	Cora	PPI
节点数	3312	2708	3890
边数	4732	5429	76584
类别数	6	7	50
特征	3703	1433	—

对于每个数据集，分别保留一定比率(10%，30%，50%)节点的标签，再去掉不可见类的标签，余下为可见类别的带标签节点。使用 SECT 算法得到节点的向量表示。对于算法性能，本文采用 Mirco-F_1 作为评价标准。

$$Micro\text{-}F_1=\frac{\sum_{i=1}^{|C|}2TP^i}{\sum_{i=1}^{|C|}(2TP^i+FP^i+FN^i)} \tag{26}$$

其中，$|C|$ 是类别的个数，如 Cora 数据集有 7 个类别，TP^i 表示在第 i 类中被判定为正类的个数，同理 FN^i 表示在第 i 类中被判定为负类的个数。

4.2 实验参数设置

在不完全平衡标签数据集上做节点分类任务，需要将数据集划分为可见标签部分及不可见标签部分。由于 3 个数据集所含类别及节点个数不同并且差异较大，对它们分别进行处理。其中，Cora 与 Citeseer 类别较少，划分两个标签为不可见，各有 C_7^2 与 C_6^2 种划分，随机划分 21 次及 15 次并进行相应的实验。而 PPI 类别较多，划分 5 个标签为不可见，随机选取 5 个不可见标签重复 20 次实验。

GAT 模型使用文献[17]中的参数，在 Cora 和 Citeseer 上使用两层注意力网络，注意力头数 $K=8$，L_2 正则化系数 $\lambda=0.0005$，每层的 dropout 为 0.6。在 PPI 数据集上，使用 3 层注意力网络，前两层 $K=4$，第三层 $K=6$，激活函数为 logistic sigmoid。GCN，APPNP，RECT 这 3 个模型使用文献[10，13-14]提供的代码及默认参数设置，嵌入向量为 200 维。

SECT 使用 SVD 将节点属性降维为 200 维的表示向量用于后续任务，语义信息模块使用 3 层注意力网络和一个 MLP 层，注意力头数 $K=4$，激活函数为 log softmax。

所有模型都训练 100 个 epoch，使用 Adam SGD optimizer 训练，学习率 $lr=0.001$。

4.3 实验结果与分析

实验基于 Cora，Citeseer，PPI 这 3 个真实的数据集进行预处理，将节点划分为可见类别与不可见类别，抽取 10%～50%的节点为训练集，将余下的节点作为测试集。预测的节点类别与真实节点标签做对比，使用 MircoF1 分类性能指标评估性能，应用不同算法的具体实验结果如表 2 所列，其中标粗的是最好结果。准确率指标在 Cora 和 Citeseer 数据集上的结果如图 5、图 6 所示，纵坐标表示准确率，横坐标表示数据集中有标签的节点所占百分比，该部分节点集即训练集。

表 2 3 个数据集上 5 种算法的 Mirco-F1 值对比

Table 2 Mirco-F1 of five algorithms classification results on three datasets

模型方法		GAT	GCN	APPNP	$SECT_{sematic}$	$SECT_{structure}$	RECT	SECT
Cora	10%	0.6472	0.6436	0.7033	0.7747	0.7762	0.8197	**0.8313**
	30%	0.6858	0.6696	0.7379	0.7912	0.8187	0.8561	**0.8663**
	50%	0.7249	0.6786	0.7602	0.8269	0.8393	0.8615	**0.8746**
Citeseer	10%	0.5082	0.4629	0.5902	0.6748	0.7011	0.7083	**0.7238**
	30%	0.6097	0.5296	0.6258	0.7072	0.7356	0.7403	**0.7489**
	50%	0.6582	0.5989	0.6409	0.7214	0.7395	0.7475	**0.7642**
PPI	10%	0.0815	0.0469	0.0428	0.1561	0.1579	0.1659	**0.1698**
	30%	0.1071	0.0449	0.0437	0.1793	0.1826	0.1956	**0.2118**
	50%	0.1367	0.0438	0.0456	0.1903	0.1941	0.2056	**0.2253**

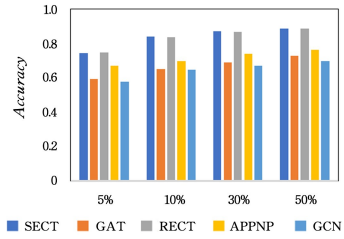

图 3 Cora 数据集上 5 种算法的准确率

Fig. 3 Accuracy of 5 algorithms on Cora

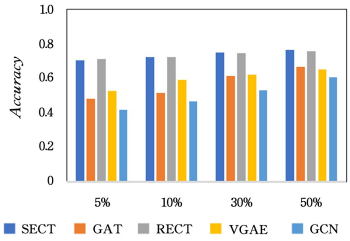

图 4 Citeseer 数据集上 5 种算法的准确率

Fig. 4 Accuracy of 5 algorithms on Citeseer

由实验结果可知，3 个数据集上 SECT 算法的分类效果均优于其他算法，与次优算法相比在 Cora 数据集上提高了 1.02%1.31%，在 Citeseer 上提升了 0.86%1.67%，在 PPI 上提升了 0.29%1.97%。

在聚合邻域信息中，GAT 与 APPNP 均考虑了邻居节点对中心节点的贡献度，有助于表示向量增大类间距离并缩小类内距离，因此分类结果优于 GCN。由于语义信息为算法提供了额外的监督信息，因此类别缺失问题上 RECT 与 SECT 的表现较好。首先，SECT 考虑了结构信息的分布，使泛化误差更小；其次，由于语义信息即为类属性分布，SECT 在优化邻域聚合时，充分利用了属性信息学习聚合权重，较在聚合时仅考虑结构影响的 RECT 效果更好。不同的网络权值的影响体现在语义信息嵌入上，其在聚合节点的邻域时将节点特征输入注意力层，学习得到节点与各邻居节点间的注意力权重，此权重用于聚合中心节点的各邻居节点信息，再有监督地学习语义信息，损失函数如式(25)所示。

由图 3 和图 4 可知，在训练集较大时，SECT 算法的分类准确率取得了最好的结果。而在 5%标签率时 SECT 的准确率不如 RECT，这是由于定义 2 中假设语义信息分布为高斯分布，而在小样本情况下的方差与整体方差存在较显著的差异，因此小样本更适合用 t 分布描述[23]，这是未来的一个研究方向。

为了更直观地显示算法的性能，使用 t-SNE 包将 3 种算法学习的节点向量映射到二维空间做可视化。节点类别以不同颜色加以区分，Case_based 和 Genetic_Algorithms 为不可见类别，结果如图 5 所示。

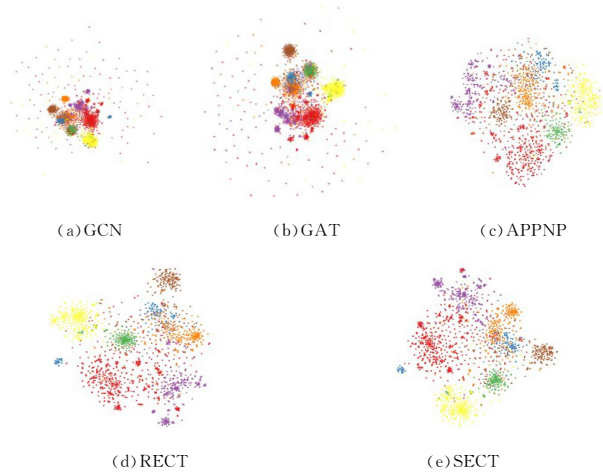

图 5 Cora 数据集上可见类别为 5 时的可视化对比

(电子版为彩图)

Fig. 5 Visualization results on Cora with visible category 5

可见与基准算法相比，SECT 算法学习到的节点向量扩大了类间距离，同时类簇内部也更加紧凑，能够清晰地区分类别边界。

4.4 结构信息学习模块的经验误差实验

针对 3.1 节中的泛化风险分析，进行经验风险的实验对比。采用与 4.1 节相同的实验条件，分别测试 3 个数据集上保留 10%的标签的经验误差，所得结果如表 3 所列，由结果可知，VGAE 的经验误差较小，在分类任务中会有较好的表现。

表 3 经验误差对比

Table 3 Comparsion of empirical error

	Cora	Citeseer	PPI
GCN	0.16213	0.26504	0.06731
VGAE	0.15608	0.24765	0.06578

4.5 语义信息学习模块的对比实验

在 SECT 语义通道与 RECT 语义通道上进行语义信息学习实验，取 10 次实验的 Mirco-F1 均值，得到的对比结果如表 4 所列。由实验结果可印证，SECT 中的注意力层可充分利用属性信息学习聚合权重，较在聚合时仅考虑结构影响的 RECT 效果更好。

表 4　两种嵌入方法语义信息学习的 Mirco-F1 值

Table 4　Mirco-F1 of sematic information learning using two embedding methods

	Cora	Citeseer	PPI
$RECT_{semantic}$	0.7304	0.6530	0.1456
$SECT_{semantic}$	0.7813	0.6796	0.1519

4.6　参数敏感性分析

为了分析 SECT 算法的参数敏感性，本文对可见类别数量、注意力头数取不同值，并基于 Citeseer 做对比实验，其他数据集上的结果类似。下面分析改动可见类别数量、注意力头数对算法性能的影响。

在 Citeseer 数据集上分别保留 10% 与 50% 的标签的情况下，改变可见类别的数量后，得到的分类性能指标 Micro-F1 值的比较情况如图 6、图 7 所示。

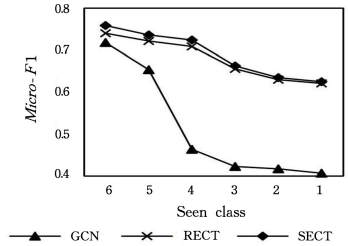

图 6　可见类别数量影响分析(10%标签)

Fig. 6　Influence analysis of number of visible category(10% labels)

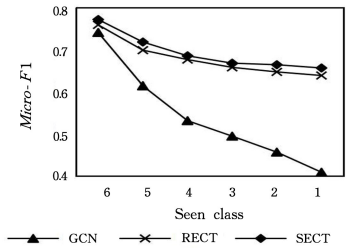

图 7　可见类别数量影响分析(50%标签)

Fig. 7　Influence analysis of number of visible category(50% labels)

当可见类别数量(seen class)减少时，GCN 性能变化较大；RECT 得益于语义额外监督信息的作用，在可见类别减少时性能浮动不大；而更好地保留语义信息的 SECT 性能浮动更小，效果最好。

下面分析注意力头数对性能的影响。在 Citeseer 数据集上保留 30%的标签率，5 个可见类别条件下改变注意力头数，分类性能指标 Mirco-F1 值的变化情况如图 8 所示。

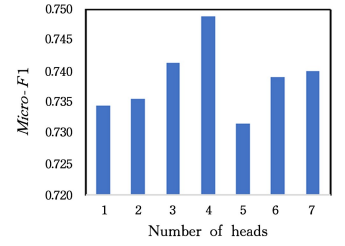

图 8　注意力头数的影响分析

Fig. 8　Influence analysis of attention heads

当头数小于 4 时，性能随头数上升，而头数大于 4 时性能逐渐下降。这表明在头数设置为 4 时，SECT 模型对标签类别语义信息的提取能力最佳。在 Citeseer 及 PPI 上也有类似的效果。

5　算法复杂度分析

本节分析并对比算法的时间复杂度及空间复杂度。设节点数为 N，属性维度为 F，边数为 E，嵌入维度为 F'，第 i 层网络的输入输出维度为 D_i^{in} 和 D_i^{out}，网络层数为 m。那么 GCN 网络的时间复杂度为 $O(N\sum_1^m D_i^{\text{in}}D_i^{\text{out}})$。含多个 GCN 层的 RECT 模型的时间复杂度可表示为 $O(N\sum_1^m D_i^{\text{in}}D_i^{\text{out}})$。在 SECT 中由于使用注意力层来计算聚合权重，增加的时间开销为 $O(E\sum_1^m D_i^{\text{out}})$，因此含图卷积和注意力层的 SECT 模型的时间复杂度为 $O(N\sum_1^m D_i^{\text{in}}D_i^{\text{out}}+E\sum_1^m D_i^{\text{out}})$。SECT 与 RECT 基准算法均采用双通道 GNN 结构，SECT 中每通道含 3 层神经网络，因此较 2 层网络的基准模型空间复杂度增加了 0.5 倍。在实验过程中，记录了算法的运行时间(单位为 s)，取 3 次实验的平均值作为结果，对比情况如表 5 所列。

表 5　运行时间对比

Table 5　Comparison of running time

	Cora	Citeseer	PPI
RECT	33.66	79.03	246.05
SECT	60.17	110.02	293.54
Increase Time	26.51	30.99	47.49

由于 SECT 中网络层数增加，在注意力层多了学习注意力权重的过程，因此较基准算法 SECT 运行时间有所增加。通过分析可知，邻域聚合的时间复杂度为 $O(N^2)$，而权重学习的复杂度大体为 $O(N)$，随着数据规模增大，权重学习对运行时间的影响会变小。由实验可知，对于数据规模较小的数据集 Cora 和 Citeseer，运行时间增加较明显；而对于数据规模较大的数据集 PPI，时间增加较不明显。

结束语　本文针对类别缺失不平衡网络嵌入过程中忽略语义特征与属性特征间的内在关系，仅依据拓扑结构提取语义特征向量的问题做出了进一步的探索。在研究了标签语义信息的内涵后，设计了更合理的语义特征提取方法，提出了融合属性特征与结构特征的语义信息增强学习算法 SECT。该模型保留了 RECT 的优势，通过加入注意力机制有效改善了学习过程中的邻域聚合操作，使嵌入向量可更好地融合语义信息。实验结果显示，该改进算法的表现优于 GNN 类算法，有助于提升在完全不平衡标签情况下的节点分类效果。在训练集为小样本情况时所提算法的表现不太理想，因为在小样本中更适用于 t 分布采样。在未来工作中，首先应该进一步发掘有价值的信息作为新的监督条件来学习更优的节点表示向量，如在属性空间加入聚类、为类别缺失节点学习置信度标签等；其次，小样本问题在现实应用中普遍存在，这为表示学习提出了新的挑战。接下来的工作希望提升小样本数据情况下的表示性能。

参 考 文 献

[1]　CUI P，WANG X，PEI J，et al. A survey on network embedding

[J]. IEEE Transactions on Knowledge and Data Engineering, 2019, 31(5): 833-852.
[2] YIN Y, JI L X, HUANG R Y, et al. Research and development of network representation learning [J]. Chinese Journal of Network and Information Security, 2019, 5(2): 77-87.
[3] BALASUBRAMANIAN M, SCHWARTZ E L. The isomap algorithm and topological stability[J]. Science, 2002, 295(5552): 7.
[4] ROWEIS S T, SAUL L K. Nonlinear dimensionality reduction by locally linear embedding [J]. Science, 2000: 290(5500): 2323-2326.
[5] BELKIN M, NIYOGI P. Laplacian eigenmaps and spectral techniques for embedding and clustering[C]// Proceedings of the 2001 14th International Conference on Neural Information Processing Systems: Natural and Synthetic. Cambridge, MA: MIT Press, 2001: 585-591.
[6] PEROZZI B, ALRFOU 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: ACM, 2014: 701-710.
[7] 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: ACM, 2016: 855-864.
[8] 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: ACM, 2015: 1067-1077.
[9] 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: ACM, 2015: 891-900.
[10] KIPF T N, WELLING M. Semi-supervised classification with graph convolutional networks[C]// International Conference on Learning Representations(ICLR). 2017.
[11] HAMILTON W L, YING R, LESKOVEC J. Inductive representation learning on large graphs[C]// Neural Information Processing Systems(NIPS). 2017: 1024-1034.
[12] PETAR V, GUILLEM C, ARANTXA C, et al. Graph attention networks [C]// Proceedings of the 6th International Conference on Learning Representations. Vancouver, BC: Elsevier, 2018: 1-12.
[13] KLICPERA J, BOJCHEVSKI A, GUNNEMANN S. Predict then propagate: Graph neural networks meet personalized pagerank[C]// International Conference on Learning Representations. 2019.
[14] WANG Z, YE X J, WANG C K, et al. Network Embedding with Completely-imbalanced Labels[J]. IEEE Transactions on Knowledge and Data Engineering(TKDE), 2020, 33(11): 3634-3647.
[15] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[C]// International Conference on Learning Representations(ICLR). 2013.
[16] MNIH A, HINTON G E. A scalable hierarchical distributed language model[C]// Advances in Neural Information Processing Systems. 2009: 1081-1088.
[17] MORIN F, BENGIO Y. Hierarchical probabilisticneural network language model[C]// Proceedings of the International Workshop on Artificial Intelligence and Statistics. 2005: 246-252.
[18] YANG C, LIU Z Y, ZHAO D L, et al. Network representation learning with rich text information[C]// Proceedings of IJCAI. 2015.
[19] KIPF T N, WELLING M. Variational graph auto-encoders [C]// NIPS Workshop on Bayesian Deep Learning. 2016.
[20] KINGMA D P, WELLING M. Auto-encoding variational bayes [C]// Proceedings of the International Conference on Learning Representations(ICLR). 2014.
[21] BOUSQUETO, ELISSEEFF A. Stability and generalization [J]. Journal of Machine Learning Research, 2002, 2(Mar): 499-526.
[22] ZHOU Z H, WANG W, GAO W, et al. Introduction to the theory of Machine Learning [M]. Beijing: China Machie Press (CMP), 2020: 92-94.
[23] STUDENT. The Probable Error of a Mean[J]. Biometrika, 1908, 6(1): 1-25.

FU Kun, born in 1979, Ph.D, associate professor. Her main research interests include social network analysis and network representation learning.

(责任编辑:喻藜)