

## 一种融合节点文本属性信息的网络表示学习算法

刘正铭, 马宏, 刘树新, 杨奕卓, 李星

(国家数字交换系统工程技术研究中心, 郑州 450002)

**摘要:** 现有网络表示学习算法主要针对网络结构信息进行表示学习, 而忽略现实网络中丰富的节点文本属性信息。为有效融合网络结构信息和节点文本属性信息进行表示学习, 提出一种新的网络表示学习算法。为实现两方面信息在训练过程中的相互约束, 建立基于参数共享的共耦神经网络训练模型, 并利用负采样和随机梯度下降的优化策略实现训练过程的快速收敛。实验结果表明, 与 Doc2Vec 算法、DeepWalk 算法、DW + D2V 算法和 TADW 算法相比, 该算法的分类性能更好。

**关键词:** 复杂网络; 网络表示学习; 信息融合; 文本属性信息; 神经网络

**中文引用格式:** 刘正铭, 马宏, 刘树新, 等. 一种融合节点文本属性信息的网络表示学习算法[J]. 计算机工程, 2018, 44(11): 165-171.

**英文引用格式:** LIU Zhengming, MA Hong, LIU Shuxin, et al. A network representation learning algorithm fusing with textual attribute information of nodes[J]. Computer Engineering, 2018, 44(11): 165-171.

## A Network Representation Learning Algorithm Fusing with Textual Attribute Information of Nodes

LIU Zhengming, MA Hong, LIU Shuxin, YANG Yizhuo, LI Xing

(National Digital Switching System Engineering and Technological R&D Center, Zhengzhou 450002, China)

**[Abstract]** The existing network representation learning algorithms mainly focus on how to represent the network structure information, and ignore the abundant textual attribute information of nodes in real network. In order to incorporate network structure information and nodes' textual attribute information, this paper presents a novel network representation learning algorithm incorporating with nodes' textual attribute information. As to achieve mutual restraint of the two part of network information during the training process, this algorithm constructs a coupled neural network training model based on parameter sharing stratagem. It applies optimization strategy based on negative sample and stochastic gradient descent to achieve rapid convergence of the training process, and performs an experimental evaluation of node classification. Experimental results demonstrate that compared with Doc2Vec algorithm, DeepWalk algorithm, DW + D2V algorithm and TADW algorithm, the classification performance of the proposed algorithm is better.

**[Key words]** complex network; network representation learning; information fusion; textual attribute information; neural network

**DOI:** 10.19678/j.issn.1000-3428.0050760

### 0 概述

近年来,随着以智能终端和社交媒体为代表的各种信息渠道的出现,大数据分析技术越来越受到人们的重视<sup>[1]</sup>。社交网络、科学引文网络等复杂网络的规模不断扩大,网络数据类型复杂多样。现实网络数据的高维性、稀疏性和异质性等特点,对现有网络分析技术带来严重挑战,这使得对于网络数据的表示学习研究具有重要意义。

网络表示学习旨在将每个网络节点映射为一个低维空间的稠密向量,使得相似的网络节点在低维空间距离较近。网络表示学习通过对网络数据形式进行变换,一方面使其包含的数据信息能够更加容易提取和分析,即由人为的特征工程转化为机器的自动特征提取,另一方面有效缓解了网络数据表示的高维性、稀疏性等问题。

传统的网络表示学习模型主要是基于特定网络关系矩阵降维得到节点的向量表示<sup>[2-5]</sup>,其复杂度通

**基金项目:** 国家自然科学基金(61521003)。

**作者简介:** 刘正铭(1995—),男,硕士研究生,主研方向为网络学习、网络信息挖掘;马宏,研究员;刘树新,助理研究员、博士;杨奕卓,硕士研究生;李星,助理研究员、博士研究生。

**收稿日期:** 2018-03-13 **修回日期:** 2018-05-03 **E-mail:** liuzhengming\_wy@163.com

常是网络节点数量的二次方,同时难以融合网络节点文本属性等异质信息进行表示学习。近年来,大量研究者开始研究基于深度学习的网络表示学习方法<sup>[6-7]</sup>。文献[8]提出了 DeepWalk 算法,通过随机游走产生节点序列,并将节点序列看作特殊的“句子”作为 Word2Vec 算法<sup>[9]</sup>输入,学习节点的向量表示。文献[10]提出了 LINE 算法,对所有网络节点间的一阶相似性和二阶相似性进行概率建模,通过最小化该概率分布和经验分布的 KL 散度得到节点的向量表示。文献[11]提出了 Node2Vec 算法,在 DeepWalk 算法基础上,通过设定 in, out 超参数控制随机游走策略,挖掘网络结构的局部特性和全局特性。文献[12]提出了一个 LsNet2Vec 模型,针对大规模网络中的链路预测问题进行网络节点的表示学习。然而,上述方法都只利用了网络结构信息,忽略了网络节点属性信息。

现实的网络数据还包括丰富的网络节点属性信息,如科学引文网络中文献题目和摘要等信息。现有融合节点文本属性信息进行表示学习的算法主要有 TADW 算法<sup>[13]</sup>,该算法将节点文本属性信息表示矩阵嵌入矩阵分解过程中实现融合表示学习。然而该算法利用 TF-IDF<sup>[14]</sup>方法编码表示节点文本属性信息,忽略了文本中词的词序信息,难以有效挖掘深层语义信息。

针对上述方法的不足,本文提出一种融合节点文本属性信息的网络表示学习算法。首先,基于 DeepWalk 思想,将网络节点结构信息的表示学习问题转化为词的表示学习问题。其次,针对节点文本属性信息的表示学习问题,利用神经网络语言模型挖掘节点文本属性的深层语义信息。最后,为实现两方面信息的融合表示学习,提出基于参数共享机制的共耦神经网络模型进行联合训练。

## 1 相关概念

为更好地描述所提模型及其具体算法,首先给出相关定义及符号表示。

**定义 1**(文本属性信息网络) 用  $G = (V, E, C)$  表示文本属性信息网络,  $V = \{v_1, v_2, \dots, v_N\}$  表示节点集合,  $N = |V|$  表示网络中的节点数量,  $E$  表示  $V$  中任意 2 个节点链接构成的集合  $E = \{e_{ij} | i, j = 1, 2, \dots, N\}$ ,  $e_{ij}$  表示节点间的链接关系紧密程度,即链接权重,  $C = \{c_1, c_2, \dots, c_N\}$ ,  $c_i$  表示与节点  $v_i$  相关联的节点文本属性信息。

**定义 2**(网络表示学习) 给定文本属性信息网络  $G = (V, E, C)$ , 网络表示学习旨在将网络  $G$  中的每一个节点  $v_i$  映射为一个低维稠密的特征向量表示  $\phi(v_i) \in \mathbb{R}^d$ , 也称为表示向量, 其中  $d \ll |V|$ 。低维空间的表示向量相似性对应于原始网络节点的相似性。

这里考虑网络节点相似性主要通过网络结构信息和网络节点文本属性信息进行刻画。也就是说在网络表示学习过程中,需要同时注意网络节点结构信息相似性保留和文本属性信息相似性保留,得到综合两方面信息的节点表示向量。

节点的表示向量  $\phi(v)$  可以看作节点  $v$  的特征向量,可直接将其作为机器学习算法的输入用于后续网络分析任务,如节点分类、链路预测等。由于表示学习过程并不涉及具体网络分析任务,因此算法所得的表示向量具有广泛适用性。

## 2 算法实现

本节首先分别介绍刻画节点文本属性信息相似性和网络结构信息相似性的基础模型,然后基于这 2 种基础模型给出融合训练模型及其算法的优化求解过程,最后结合算法伪代码进行算法复杂度分析。

### 2.1 基础模型

#### 2.1.1 节点文本属性信息表示学习

近年来,基于 CBOW<sup>[9]</sup>神经网络语言模型的词向量表示学习方法,通过窗口上下文预测中间词,较好地保留了文本语句中的词序信息。在此基础上,文献[15]提出了用于文本向量表示的 Doc2Vec 算法,在很多应用中取得了较好的结果。因此,将其作为本文融合算法的基础模型之一。

如图 1 所示,对于任意词  $w$ , 给定左右窗口大小为  $b$  的上下文词集合  $context(w) = \{w_{-b}, w_b\}$ ,  $v(w)$  表示一个从词  $w$  到对应节点的映射函数,矩阵  $W$  中的每一行表示一个词对应的表示向量,矩阵  $U_w$  中的每一行表示一个节点对应的文本属性信息的表示向量。

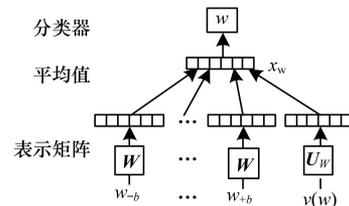


图 1 节点文本属性信息表示学习模型

算法基本思想是在已知上下文  $context(w)$  和  $v(w)$  的情况下,预测到词  $w$  的概率最大。其对应最大化目标函数如下:

$$L_A = \log_a \prod_{w \in D} p(w | context(w), v(w)) = \sum_{w \in D} \log_a p(w | context(w), v(w)) \quad (1)$$

其中,  $D$  对应于节点文本属性信息中所有词的集合。  $p(w | context(w), v(w))$  定义为如下 Softmax 函数:

$$p(w | context(w), v(w)) = \frac{\exp(\mathbf{x}_w^T \cdot \mathbf{v}'(w))}{\sum_{u \in D} \exp(\mathbf{x}_w^T \cdot \mathbf{v}'(u))} \quad (2)$$

其中,  $\mathbf{v}(u)$  和  $\mathbf{v}'(u)$  表示  $u$  的表示向量及其辅助向量,  $\mathbf{x}_w$  采用累加求和的形式计算如下:

$$\mathbf{x}_w^T = \sum_{u \in \{v(w)\} \cup \text{context}(w)} \mathbf{v}(u) \quad (3)$$

通过模型训练后,  $U_w$  将作为最后所有节点的文本属性信息表示向量矩阵输出。

### 2.1.2 节点网络结构信息表示学习

对于网络结构信息表示学习问题, 主要分为采样和训练 2 个阶段。在采样阶段, 使用文献 [8] 提出的随机游走策略捕捉网络结构信息。从任意节点  $v_i$  出发, 随机游走固定长度  $l$  得到随机游走序列  $S = \{v_i, v_{i+1}, v_{i+2}, \dots, v_{i+l}\}$  作为训练集。在训练阶段, 将随机游走序列看作特殊的“句子”, 作为 CBOW 模型<sup>[10]</sup> 的输入, 学习节点向量表示。如图 2 所示, 对于任意节点  $v$ , 假设给定左右窗口大小为  $b$  的上下文节点集合为  $\text{context}(v) = \{v_{-b}:v_b\}$ , 矩阵  $U_s$  中的每一行表示一个节点对应的网络结构信息的表示向量。

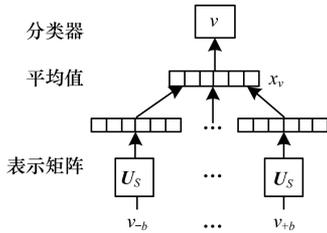


图 2 网络结构信息表示学习模型

与第 2.1.1 节类似, 在已知上下文  $\text{context}(v)$  的情况下, 预测到节点  $v$  的概率最大, 其对应最大化目标函数为:

$$L_s = \log_a \prod_{v \in V} p(v | \text{context}(v)) = \sum_{v \in V} \log_a p(v | \text{context}(v)) \quad (4)$$

其中,  $V$  是所有节点的集合。  $p(v | \text{context}(v))$  定义为如下 Softmax 函数:

$$p(v | \text{context}(v)) = \frac{\exp(\mathbf{x}_v^T \cdot \mathbf{v}'(v))}{\sum_{u \in V} \exp(\mathbf{x}_v^T \cdot \mathbf{v}'(u))} \quad (5)$$

这里  $\mathbf{x}_v$  采用累加求和的形式计算如下:

$$\mathbf{x}_v^T = \sum_{u \in \text{context}(v)} \mathbf{v}(u) \quad (6)$$

通过模型训练后,  $U_s$  将作为最后所有节点的网络结构信息表示向量矩阵输出。

### 2.2 融合表示学习模型及其算法优化

为实现节点网络结构信息和文本属性信息的融合表示, 最简单的方法就是拼接。如图 3(a) 所示, 记通过文本属性信息表示学习模型训练得到的表示矩阵为  $U_w$ , 通过网络结构信息表示学习模型训练得到的表示矩阵为  $U_s$ , 直接拼接得到最终的节点表示向量矩阵  $U_+$ , 即  $U_+ = U_w \oplus U_s$ , 然而这种方法由于  $U_w$  和  $U_s$  在训练过程中相互独立, 属于训练后结合, 缺少了两方面信息在训练过程中的相互补充与制约。因此, 提出基于参数共享的交叉训练机制实现融合表示学习, 如图 3(b) 所示。首先, 使用融合表示向量矩阵  $U$  替换基础模型中的  $U_w$  和  $U_s$ , 建立耦合神经网络模型, 如图 4 所示。

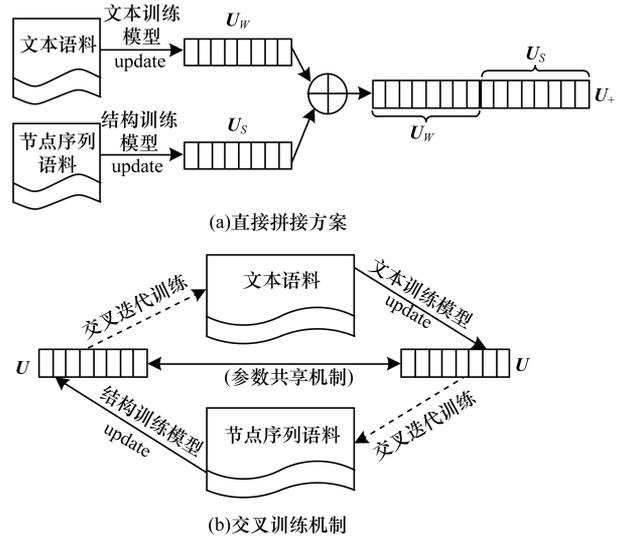


图 3 2 种节点文本属性的融合方案

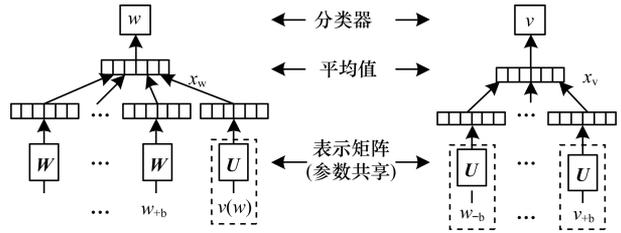


图 4 融合节点文本属性信息的表示学习模型

左右两部分的表示学习模型交替训练,  $U$  由 2 个模型共享, 即  $U$  在训练过程中相互传递。最后, 通过反复迭代, 得到融合两方面信息的节点向量表示, 其对应的最大化目标函数为:

$$L = L_A + L_s = \sum_{v \in V} \log_a p(v | \text{context}(v)) + \sum_{w \in D} \log_a p(w | \text{context}(w), v(w)) \quad (7)$$

其直观解释是: 一方面融合表示向量和上下文词向量一起用于预测中间词  $w$ , 使得融合表示向量包含节点文本属性信息; 另一方面融合表示向量又参与节点网络结构信息的表示学习训练, 通过节点网络结构信息修正融合表示向量。在反复迭代过程中, 实现两方面信息的相互补充与制约。

采用随机梯度上升方法进行迭代训练, 考虑到计算式 (2) 和式 (5) 时需要分别遍历整个词集合与节点集合, 不适合在大规模网络的实际应用, 文献 [16] 提出了基于负采样 (Negative Sampling, NEG) 的优化策略用于降低计算复杂度, 给出式 (5) 的近似表示如下:

$$g(v) = \prod_{u \in \{v\} \cup \text{NEG}(v)} [\sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u))]^{L^v(u)} \cdot [1 - \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u))]^{1 - L^v(u)} \quad (8)$$

其中,  $L^v(u)$  为 0-1 判决函数, 当  $u = v$  时,  $L^v(u) = 1$ , 否则  $L^v(u) = 0$ ,  $\sigma(x) = 1 / (1 + e^{-x})$ 。  $\text{NEG}(v)$  表示正样本  $(v, \text{context}(v))$  对应的负样本集。从式 (8) 不难看出, 负采样的基本思想是最大化正样本出现概率的同时最小化负样本出现概率。

下面进一步推导表示向量的更新公式,将式(8)带入式(4)中可得:

$$L_S = \sum_{v \in Vu \in \{v\} \cup NEG(v)} \{L^v(u) \cdot \log_a \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u)) + [1 - L^v(u)] \cdot \log_a [1 - \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u))]\} \quad (9)$$

为求导方便,记式(9)两次求和项如下:

$$L_S(v, u) = L^v(u) \cdot \log_a \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u)) + [1 - L^v(u)] \cdot \log_a [1 - \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u))] \quad (10)$$

首先考虑  $L_S(v, u)$  关于  $\mathbf{v}'(u)$  的梯度,推导如下:

$$\begin{aligned} \frac{\partial L_S(v, u)}{\partial \mathbf{v}'(u)} &= \frac{\partial}{\partial \mathbf{v}'(u)} \{L^v(u) \cdot \log_a \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u)) + \\ & [1 - L^v(u)] \cdot \log_a [1 - \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u))]\} = \\ & L^v(u) \cdot [1 - \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u))] \cdot \\ & \mathbf{x}_v - [1 - L^v(u)] \cdot \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u)) \cdot \mathbf{x}_v = \\ & [L^v(u) - \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u))] \cdot \mathbf{x}_v \quad (11) \end{aligned}$$

同理,可求出  $L_S(v, u)$  关于  $\mathbf{x}_v$  的梯度如下:

$$\frac{\partial L_S(v, u)}{\partial \mathbf{x}_v} = [L^v(u) - \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u))] \cdot \mathbf{v}'(u) \quad (12)$$

由式(11)和式(12)可得节点表示向量  $\mathbf{v}(\tilde{v})$  和辅助向量  $\mathbf{v}'(u)$  的更新公式如下:

$$\begin{aligned} \mathbf{v}(\tilde{v}) &= \mathbf{v}(\tilde{v}) + \eta \sum_{u \in \{v\} \cup NEG(v)} [L^v(u) - \\ & \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u))] \cdot \mathbf{v}'(u), \\ & \tilde{v} \in context(v) \quad (13) \end{aligned}$$

$$\mathbf{v}'(u) = \mathbf{v}'(u) + \eta [L^v(u) - \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u))] \cdot \mathbf{x}_v, \quad u \in \{v\} \cup NEG(v) \quad (14)$$

对于节点文本属性信息表示学习模型的计算方法类似,在此不再赘述,直接给出最后的更新公式。

$$\begin{aligned} \mathbf{v}(\tilde{w}) &= \mathbf{v}(\tilde{w}) + \eta \sum_{u \in \{w\} \cup NEG(w)} [L^w(u) - \\ & \sigma(\mathbf{x}_w^T \cdot \mathbf{v}'(u))] \cdot \mathbf{v}'(u), \\ & \tilde{w} \in \{v(w)\} \cup context(w) \quad (15) \end{aligned}$$

$$\mathbf{v}'(u) = \mathbf{v}'(u) + \eta [L^w(u) - \sigma(\mathbf{x}_w^T \cdot \mathbf{v}'(u))] \cdot \mathbf{x}_w, \quad u \in \{w\} \cup NEG(w) \quad (16)$$

### 2.3 融合算法流程及其复杂度分析

融合算法伪代码如下:

**算法1** 融合节点文本属性信息的网络表示学习算法

**输入** 信息网络  $G = (V, E, C)$ , 迭代次数  $r$ , 表示向量维度  $d$ , 采样窗口左右大小  $b$ , 随机游走长度  $l$ , 随机游走次数  $r'$ , 负采样样本数  $k$

**输出** 节点融合表示向量矩阵  $U$ , 每一行对应节点表示向量  $\mathbf{v}(u)$ ,  $u \in V$

训练数据集采样步骤

1. 对于节点文本属性信息, 给定参数  $(b)$ , 以采样窗口大小  $b$  采样文本信息, 构成文本属性信息训练集  $\{(w, context(w), v(w), NEG(w))\}$ 。

2. 对于网络结构信息, 给定参数  $(l, b, r', k)$ , 首先通过随机游走产生节点序列集合, 再以采样窗口大小  $b$  采样节点序列, 构成网络结构信息训练集  $\{(v, context(v), NEG(v))\}$ 。

迭代训练步骤如下:

3. for iter = 1 to  $r$

4. for  $w$  in  $D$

5. random sample  $(w, context(w), v(w), NEG(w))$

6. update = 0

7.  $WTHZ | \mathbf{x}_w^T = \sum_{u \in \{v(w)\} \cup context(w)} \mathbf{v}(u)$

8. for  $u$  in  $\{w\} \cup NEG(w)$

9. delta =  $\eta [L^w(u) - \sigma(\mathbf{x}_w^T \cdot \mathbf{v}'(u))]$

10. update = update + delta  $\cdot \mathbf{v}'(u)$

11.  $\mathbf{v}'(u) = \mathbf{v}'(u) + \text{delta} \cdot \mathbf{x}_w$  // 辅助向量更新  
end

12. for  $u$  in  $\{v(w)\} \cup context(w)$

13.  $\mathbf{v}(u) = \mathbf{v}(u) + \text{update}$  // 表示向量更新 (词向量及节点融合表示向量)

14. end

15. end

16. for  $v$  in  $V$

17. random sample  $(v, context(v), NEG(v))$

18. update = 0

19.  $\mathbf{x}_v^T = \sum_{u \in context(v)} \mathbf{v}(u)$

20. for  $u$  in  $\{v\} \cup NEG(v)$

21. delta =  $\eta [L^v(u) - \sigma(\mathbf{x}_v^T \cdot \mathbf{v}'(u))]$

22. update = update + delta  $\cdot \mathbf{v}'(u)$

23.  $\mathbf{v}'(u) = \mathbf{v}'(u) + \text{delta} \cdot \mathbf{x}_v$  // 辅助向量更新

24. end

25. for  $u$  in  $context(v)$

26.  $\mathbf{v}(u) = \mathbf{v}(u) + \text{update}$  // 表示向量更新 (节点融合表示向量)

27. end

28. end

29. end

下面结合算法伪代码(算法1)分析算法流程并讨论其复杂度问题。

首先, 对于训练数据集采样部分, 采用文献[16]提出的带权采样策略分别采样给定词  $w$  和节点  $v$  的负样本集:  $NEG(w)$  和  $NEG(v)$ 。其基本思想是: 以词  $w$  的负样本集  $NEG(w)$  为例, 从噪声分布  $P_n(w)$  中采样  $k$  个负样本。按照文献[16]的建议, 取  $k=5$ ,  $P_n(w) \propto [count(w) / \sum_{u \in D} count(u)]^{3/4}$ 。

其次, 对于迭代训练部分, 一方面使用随机梯度上升法(对应求极大值)作为优化更新策略, 式(13)~式(16)给出了向量更新公式; 另一方面基于参数共享策略进行交叉迭代训练: 步骤4~步骤15实现了节点文本属性信息的表示学习, 步骤16~步骤29实现了网络结构信息的表示学习, 由于节点融合表示向量在两部分模型中相互传递, 使得在训练过程中受到两方面信息的相互补充与制约。迭代过程中, 对于给定的词  $w$ , 在负采样

策略下,计算次数从式(3)的 $|D|$ (语料库大小)次减少到 $1+k$ 次。

最后,分析算法的整体复杂度问题。在单次迭代过程中,对于给定词 $w$ ,在负采样策略下,计算次数从式(3)的 $|D|$ (语料库大小)次减少到 $1+k$ 次。遍历词集合,计算次数为 $|D| \cdot (1+k)$ 次。同理,对于给定节点 $v$ ,遍历节点集,计算次数为 $|V| \cdot (1+k)$ 次。因此,迭代 $r$ 次后,整体计算复杂度为 $o(r \cdot (|D| + |V|) \cdot (1+k))$ 。在实际应用场景中,由于 $r, k \ll |D|, |V|$ ,因此算法计算时间复杂度和网络规模成线性比例关系,算法可扩展到大规模场景的实际应用。

### 3 实验验证与分析

为验证本文提出算法的有效性,在 2 个公开数据集上与具有代表性的表示学习算法进行对比。

#### 3.1 实验数据集

DBLP 数据集来源于 AMiner 网站公开数据集。本文抽取其中 4 个知名国际会议论文数据(CIKM, KDD, IJCAI, CVPR),将论文作为网络节点,标题信息作为节点文本属性信息,利用引用关系构建引文网络,包含节点 18 223 个,连边 15 867 条,4 类节点标签对应不同的会议论文集。

CiteSeer-M10 数据集来源于 CiteSeerX 网站中抽取的数据集。本文将文献[17]从该网站中抽取的包含 10 个方向论文引用关系的数据集作为实验数据集。将论文作为网络节点,标题信息作为节点文本属性信息,利用引用关系构建引文网络,包含节点 10 310 个,连边 77 218 条,10 类节点标签对应不同方向的论文集。

#### 3.2 对比算法

将对比较算法分为 3 类:1)仅利用节点文本属性信息;2)仅利用网络结构信息;3)同时利用两方面信息的融合算法。

下面简要介绍对比算法:

1) Doc2Vec 算法:仅利用节点文本属性信息进行表示学习。

2) DeepWalk 算法:仅利用网络结构信息进行表示学习。

3) DW + D2V 算法:将 Doc2Vec 算法和 DeepWalk 算法学习的表示向量进行拼接,使得到的节点表示向量既包含文本属性信息又包含网络结构信息。

4) TADW 算法:通过矩阵分解的形式,直接利用节点文本属性信息和网络结构信息得到节点表示向量。

本文算法的主要参数设定为表示向量维度 $d = 200$ ,迭代次数 $r = 10$ ,其余参数设定为对应子结构的原始文献给出的建议值:文献[15]中的

Doc2Vec 算法设定文本属性信息表示学习窗口大小为 10;文献[8]根据 DeepWalk 算法对随机游走的讨论,设定游走长度 $l = 40$ ,窗口大小为 10,游走次数 $r' = 80$ 。为保持一致,各对比算法维度都设置为 $d = 200$ 。

#### 3.3 评测任务及其指标

评测方法与文献[11,13]类似,首先进行无监督的表示学习,然后将其用在多标签分类任务中,比较不同算法的性能。基本思想是具有较好标签预测能力的表示学习算法能够更加准确地从原始网络数据中提取节点特征向量表示。由于评测数据集是多分类问题,因此在评价指标选择问题上,先在各混淆矩阵上分别计算准确率和召回率,记为 $(P_1, R_1), (P_2, R_2), \dots, (P_n, R_n)$ ,再计算平均值,得到宏准确率(Macro\_P)、宏召回率(Macro\_R)及相应的宏 F 值(Macro\_F):

$$Macro\_P = \frac{1}{n} \sum_{i=1}^n P_i \quad (17)$$

$$Macro\_R = \frac{1}{n} \sum_{i=1}^n R_i \quad (18)$$

$$Macro\_F = \frac{2 \times Macro\_R \times Macro\_P}{Macro\_R + Macro\_P} \quad (19)$$

为方便进行算法比较,与文献[11,13]一致,统一采用 SVM 线性分类器进行节点分类任务,排除不同分类器对节点分类性能造成影响的情况。为考察算法在不同监督信息量情况下的标签预测性能,随机取训练集大小从 10% ~ 90%,剩余部分作为测试集,重复 10 次取结果平均值。实验流程如图 5 所示。

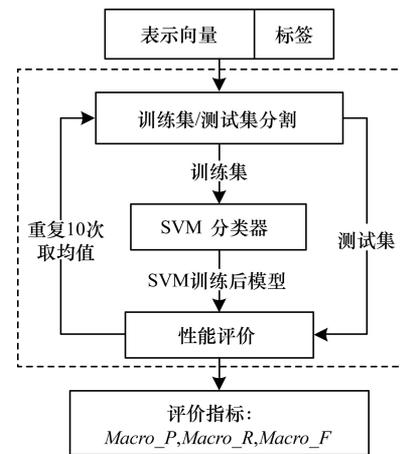


图 5 实验流程

#### 3.4 实验结果分析

图 6 和图 7 分别记录了在 DBLP 和 CiteSeer-M10 数据集上的不同训练率下(10% ~ 90%, 间隔 20% 进行测试)的 3 种节点分类性能指标结果,即宏准确率、宏召回率和宏 F 值。实验结果显示,本文所提算法的节点分类性能高于比较算法。

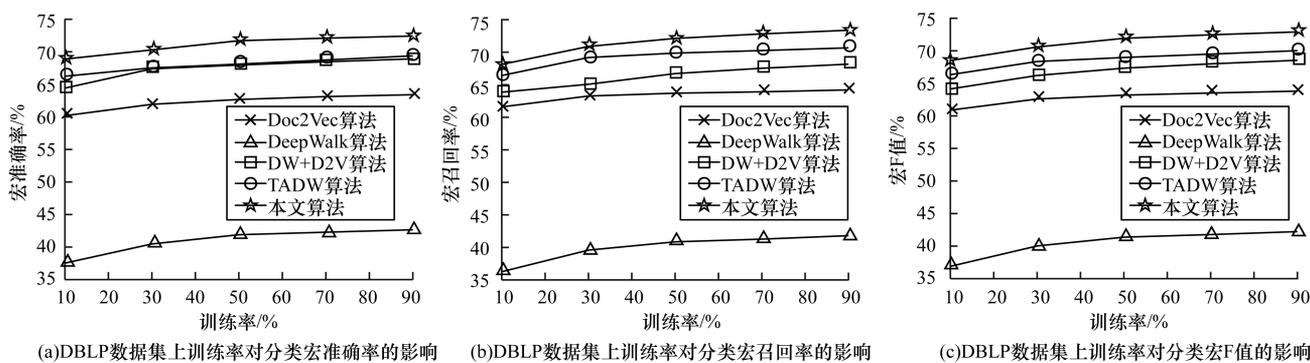


图6 DBLP数据集上的分类结果

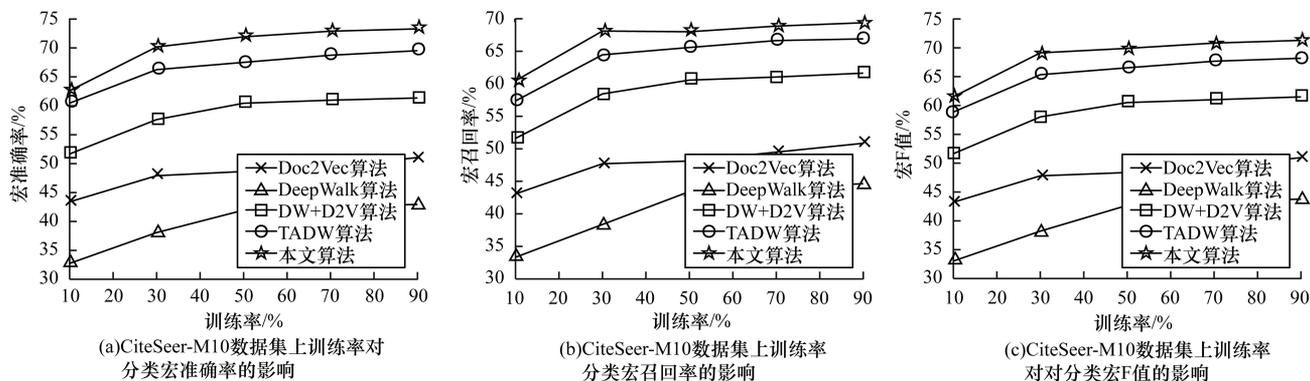


图7 CiteSeer-M10数据集上的分类结果

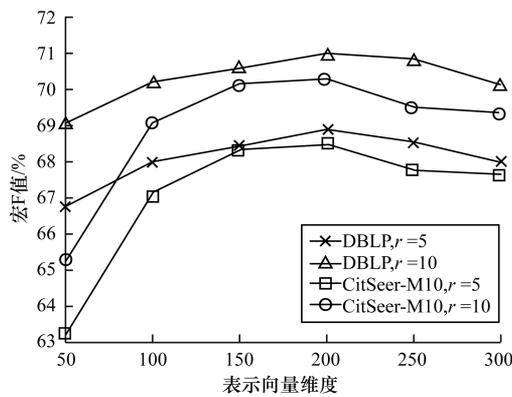
下面从两方面分析实验结果：

1) 融合算法优势明显。Doc2Vec 算法和 DeepWalk 算法分别挖掘了节点文本属性信息和结构信息,但效果都较为普通。基于简单拼接的 DW + D2V 算法性能进一步提升,但是相比于融合模型仍然有提升空间。在 30% 的训练率情况下,在 DBLP 网络中,本文算法的分类宏 F 值比 DW + D2V 算法提高了 4.3%,比融合算法 TADW 提高了 2.2%;在 CiteSeer-M10 网络中,本文算法的分类宏 F 值比 DW + D2V 算法提高了 11%,比融合算法 TADW 提高了 3.8%。

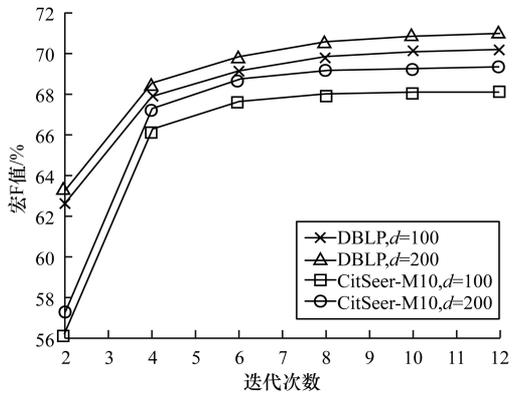
2) 神经网络特征挖掘优势明显。和通过矩阵分解方式进行融合表示的 TADW 算法相比,基于共耦神经网络的本文算法平均节点分类准确率在 DBLP 和 CiteSeer-M10 网络中分别达到 68% 和 71%,比 TADW 算法分别提高了 3% 和 3.6%。作为本文算法子结构的 Doc2Vec 文本表示学习算法仅依赖节点文本属性信息的情况下就达到较好的节点分类效果。如图 6(a) 和图 7(a) 所示,在 30% 训练率下,在 DBLP 和 CiteSeer-M10 网络上的节点分类准确率分别达到 61.9% 和 47.9%。这一方面说明结合文本属性信息的重要性,另一方面也说明了神经语言模型在挖掘文本语义信息方面的巨大优势,这也是结合神经语言模型改进网络表示学习算法的初衷。

### 3.5 算法参数敏感性分析

本文算法包含了表示向量维度  $d$  和融合算法迭代次数  $r$  这 2 个主要超参数,本节将通过实验分析超参数的选择对算法用于多标签节点分类问题性能好坏的影响。通过改变参数取值,得到不同的节点表示向量。按照图 5 的实验流程,在 30% 训练率的情况下,测试不同的节点表示向量对多标签节点分类问题性能指标宏 F 值的影响,实验结果如图 8 所示。图 8(a) 表示了改变表示向量维度  $d$  对算法分类预测性能的影响, $d$  取值从 50 ~ 300,每间隔 50 进行一次实验。随着表示向量维度的增加,分类预测宏 F 值逐渐增加,说明了较高维度能够捕获更多的网络信息,形成更具区分性的网络表示。然而同时也注意到,表示维度增加到 200 维以后,分类预测宏 F 值有所下降。这说明采用过多的表示向量维度衡量网络节点相似性,减少了具有重要区分度特征的权重影响,反而导致性能损失。因此,200 维的表示向量维度较为合适。图 8(b) 是改变算法迭代次数  $r$  对算法分类预测性能的影响,将迭代次数变化范围设置为 2 ~ 12,间隔 2 次进行一次实验。随着迭代次数的增加,分类预测宏 F 值明显提升,体现了交叉训练过程中两方面信息的相互补充。迭代次数超过 10 次以后,分类预测性能趋于稳定,说明融合模型能够挖掘的网络信息趋于稳定。因此,迭代次数超过 10 次后停止迭代更新。



(a)表示向量维度对分类宏F值的影响



(b)迭代次数对分类宏F值的影响

图8 超参数对算法分类性能指标宏F值的影响结果

#### 4 结束语

本文基于神经语言模型提出了一个结合节点文本属性信息的网络表示学习算法,实现了节点文本属性信息和网络结构信息的融合表示学习。针对文本属性信息和网络结构信息等异质信息难以有效融合表示的问题,给出基于参数共享的共耦神经网络模型用于融合训练。在2个真实世界网络数据集上的实验结果表明,该算法有效实现了融合表示学习,在面向节点分类的评测任务中,算法性能有一定提升。算法复杂度与网络规模大小成线性比例关系,能够适用于大数据时代背景下的大规模复杂信息网络的表示学习问题。然而,该算法仅考虑了节点文本属性信息,下一步将针对实际网络中存在的图像信息、语音信息等其他异质信息对算法进行优化。

#### 参考文献

[1] 李贞镐,金德鹏. 基于移动大数据的城市深夜公交线路改进方案[J]. 计算机工程,2018,44(4):23-27.  
 [2] ROWEIS S T, SAUL L K. Nonlinear dimensionality reduction by locally linear embedding [J]. Science, 2000,290(5500):2323-2326.  
 [3] BELKIN M, NIYOGI P. Laplacian eigenmaps and spectral techniques for embedding and clustering[C]// Proceedings of Advances in Neural Information

Processing Systems. British Columbia, Canada: MIT Press,2002:585-591.  
 [4] TENENBAUM J B, DE SILVA V, LANGFORD J C. A global geometric framework for nonlinear dimensionality reduction[J]. Science,2000,290(5500):2319-2323.  
 [5] SHAW B, JEBARA T. Structure preserving embedding[C]// Proceedings of the 26th Annual International Conference on Machine Learning. New York, USA: ACM Press, 2009: 937-944.  
 [6] 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.  
 [7] CAI Hongyun, ZHENG V W, CHANG K. A comprehensive survey of graph embedding: problems, techniques and applications[EB/OL]. [2018-03-13]. https://arxiv.org/abs/1709.07604.  
 [8] 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.  
 [9] MIKOLOV T, SUTSKEVER I, CHEN Kai, et al. Distributed representations of words and phrases and their compositionality[C]//Proceedings of Advances in Neural Information Processing Systems. Lake Tahoe, USA: Curran Associates,2013:3111-3119.  
 [10] TANG Jian, QU Meng, WANG Mingzhe, 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.  
 [11] GROVER A, LESKOVEC J. Node2vec; Scalable feature learning for networks [C]//Proceedings of the 22nd ACM International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM Press,2016: 855-864.  
 [12] 李志宇,梁循,周小平,等. 一种大规模网络中基于节点结构特征映射的链接预测方法[J]. 计算机学报,2016,39(10):1947-1964.  
 [13] YANG Cheng, LIU Zhiyuan, ZHAO Deli, et al. Network representation learning with rich text information[C]// Proceedings of the 24th International Joint Conference on Artificial Intelligence. Buenos Aires, Argentina: AAAI Press,2015:2111-2117.  
 [14] SPARCK J K. A statistical interpretation of term specificity and its application in retrieval[J]. Journal of Documentation,1972,28(1):11-21.  
 [15] LE Q, MIKOLOV T. Distributed representations of sentences and documents [C]//Proceedings of International Conference on Machine Learning. Beijing, China:[s. n.],2014:1188-1196.  
 [16] MIKOLOV T, SUTSKEVER I, CHEN Kai, et al. Distributed representations of words and phrases and their compositionality[C]//Proceedings of Advances in Neural Information Processing Systems. Lake Tahoe, USA: Curran Associates,2013:3111-3119.  
 [17] LIM K W, BUNTINE W. Bibliographic analysis with the citation network topic model [C]//Proceedings of the 6th Asian Conference on Machine Learning. Nha Trang, Vietnam: MLR Press,2014:142-158.