

# 基于多视图集成的网络表示学习算法

冶忠林<sup>1</sup> 赵海兴<sup>1,2</sup> 张科<sup>2</sup> 朱宇<sup>2</sup>

(陕西师范大学计算机科学学院 西安 710062)<sup>1</sup> (青海师范大学计算机学院 西宁 810008)<sup>2</sup>

**摘要** 现有的网络表示学习算法主要为基于浅层神经网络的网络表示学习和基于神经矩阵分解的网络表示学习。基于浅层神经网络的网络表示学习又被证实是分解网络结构的特征矩阵。另外,现有的大多数网络表示学习仅仅从网络的结构学习特征,即单视图的表示学习;然而,网络本身蕴含有多种视图。因此,文中提出了一种基于多视图集成的网络表示学习算法(MVENR)。该算法摒弃了神经网络的训练过程,将矩阵的信息融合和分解思想融入到网络表示学习中。另外,将网络的结构视图、连边权重视图和节点属性视图进行了有效的融合,弥补了现有网络表示学习中忽略了网络连边权重的不足,解决了基于单一视图训练时网络特征稀疏的问题。实验结果表明,所提 MVENR 算法的性能优于网络表示学习中部分常用的联合学习算法和基于结构的网络表示学习算法,是一种简单且高效的网络表示学习算法。

**关键词** 网络表示学习,网络嵌入学习,复杂网络编码学习,网络可视化,表示学习

**中图分类号** TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.01.018

## Network Representation Learning Based on Multi-view Ensemble Algorithm

YE Zhong-lin<sup>1</sup> ZHAO Hai-xing<sup>1,2</sup> ZHANG Ke<sup>2</sup> ZHU Yu<sup>2</sup>

(School of Computer Science, Shaanxi Normal University, Xi'an 710062, China)<sup>1</sup>

(College of Computer, Qinghai Normal University, Xining 810008, China)<sup>2</sup>

**Abstract** The existing network representation learning algorithms mainly consist of the methods based on the shallow neural network and the approaches based on neural matrix factorization. It has been proved that network representation learning based on shallow neural network is to factorize feature matrix of network structure. In addition, most of the existing network representation algorithms learn the features from the structure information, which is a single view representation learning for networks. However, there are various kinds of views in the network. Therefore, this paper proposed a network representation learning approach based on multi-view ensemble (MVENR). The algorithm abandons the neural network training process and integrates the idea of matrix information ensemble and factorization into the network representation vectors. MVENR gives effective combination strategy between the network structure view, The link weight view and the node attribute view. Meanwhile, it makes up the shortage of neglecting the network link weight, and solves the sparse network feature problem for using single view training. The experimental results show that the proposed algorithm outperforms the commonly joint learning algorithms and the methods purely based on network structure features, and it is a simple and efficient network representation learning algorithm.

**Keywords** Network representation learning, Network embedding learning, Complex network encoding learning, Network visualization, Representation learning

## 1 引言

网络表示学习同样是复杂网络中特征提取的重要方法,它被应用于各类机器学习任务中,如网络节点分类<sup>[1]</sup>、链路预测<sup>[2]</sup>和推荐系统<sup>[3-4]</sup>等。网络表示学习旨在对各类网络学

习得到低维度、压缩的、稠密的分布式表示向量,可形象地解释为研究如何对网络中的每个节点进行编码,使得网络中相近的节点在网络表示空间中具有更近的空间距离。

现有的网络表示学习算法主要是基于网络结构特征进行学习,即输入为网络的结构特征,学习过程为一个浅层的三层

收到日期:2018-05-05 返修日期:2018-07-16 本文受国家自然科学基金支持项目(61663041,61763041),长江学者和创新研究团队项目(IRT\_15R40),中国教育部春辉计划研究基金项目(Z2014022),青海省自然科学基金项目(2013-Z-Y17,2014-ZJ-721),中央高校基本科研业务费专项资金(2017TS045)资助。

冶忠林(1989-),男,博士生,CCF会员,主要研究方向为自然语言处理;赵海兴(1969-),男,博士,教授,主要研究方向为复杂网络、自然语言处理,E-mail:h. x. zhao@163. com(通信作者);张科(1986-),男,博士生,主要研究方向为复杂网络;朱宇(1986-),男,博士生,主要研究方向为复杂网络。

神经网络,输出为网络节点的表示向量。基于网络结构的网络表示学习算法仅基于网络的结构特征,学习得到的表示向量倾向于反映网络中节点之间的结构关联性,如共同邻居、较短的可达路径等。该类算法的典型代表为 DeepWalk<sup>[5]</sup>。另外一类网络表示学习方法为基于联合学习的网络表示学习,该类算法的基础算法同样为基于结构的网络表示学习算法,底层为一个浅层的三层神经网络,但是神经网络的输入却变为多层异构网络,而非同构网络。该类算法的典型代表为 TriDNR<sup>[6]</sup>。

表示学习受到越来越多的科研人员关注的主要原因是,语义网络中词语的表示学习获得了极大的成功,在自然语言处理的各类任务中发挥了超强的性能<sup>[7]</sup>。该领域的代表算法为 Word2Vec<sup>[8]</sup>算法。Word2Vec 通过将当前词和上下文词语对输入到浅层神经网络中,得到词语的表示向量。受 Word2Vec 算法的启发,针对大规模网络表示学习的 DeepWalk 算法<sup>[5]</sup>被提出。DeepWalk 和 Word2Vec 的底层学习算法是相同的,唯一不同的是输入的对象:Word2Vec 主要用于语言模型,输入的对象为词对;DeepWalk 通过随机游走算法将图上的随机游走序列当作语言模型中的句子,之后按照 Word2Vec 的思路训练网络表示向量。目前,DeepWalk 算法已被成功应用于很多任务中<sup>[9]</sup>。

已有研究证明,基于 Skip-Gram 模型的 Word2Vec 算法的实质为对一个目标矩阵 SPPMI<sup>[10]</sup>进行矩阵分解;同样,基于 Skip-Gram 模型的 DeepWalk 算法的实质为对一个目标矩阵  $M=(A+A^2)/2$ <sup>[11]</sup>进行矩阵分解, $A$  为节点之间的概率转移矩阵。基于矩阵分解的网络表示学习理论——MMDW<sup>[12]</sup>算法被提出,该算法采用 SVM 中的最大间隔算法对目标矩阵  $M$  分解之后得到的  $W$  矩阵进行优化处理。TADW<sup>[13]</sup>算法同样基于矩阵分解理论,但是与 MMDW 算法不同,TADW 算法使用了诱导矩阵补全算法<sup>[14]</sup>进行矩阵分解,在矩阵分解的过程中添加了节点的文本特征因子,使得学习得到的网络表示既有网络的结构特征,又有节点的文本特征。另外,还有一些基于半监督学习的网络表示学习算法,例如 TriDNR<sup>[6]</sup>,GCN<sup>[15]</sup>,DDRW<sup>[16]</sup>和 Panetoid<sup>[17]</sup>等。TriDNR 在网络表示学习模型中添加了文本特征和节点的标签信息,并使用两个神经网络训练节点的表示。TriDNR 的实质是学习网络节点内部结构、节点与词语、词语与标签之间的关系。DDRW<sup>[16]</sup>算法在网络表示学习的过程中融入了节点的类别标签信息。除此之外,还有一些社团增强和高阶逼近的表示学习算法,如 M-NMF<sup>[18]</sup>和 NEU<sup>[19]</sup>。NEU 算法实质上是一个网络表示转换等式,通过其他任意的网络表示学习算法得到低阶的网络表示,使用 NEU 高阶转换等式,低阶的网络表示就变成了高阶的网络表示。

现有的大多数网络表示学习算法仅从网络的单一视图特征学习低维度的网络表示向量,但网络本身具有多视图的特征,不同的视图反映了网络所蕴含的某个特性,且不同的视图特征对网络的重要性也不同,因此,如果将网络的多个视图进行联合学习,且对不同的视图赋予不同的权重,则可能训练出稳定、有效的网络表示向量,学习得到的表示向量可被用于各类机器学习任务中。例如,现有的社交网络数据结构非常复

杂,网络节点中含有大量的属性文本信息,以及各种各样的关联关系(Side Information)。本文将能够反映网络节点之间的关联关系称为视图。视图的概念在聚类算法中应用得非常广泛。单一视图即能够反映网络信息的某一类特征视图。本文对网络的不同视图进行相互的特征补偿学习,进而能够借助多视图集成学习算法训练出稳健而高效的网络特征表示向量。综上,受基于信息融合的矩阵分解算法的启发,本文提出了一种基于多视图集成的网络表示学习算法(MVENR)。

本文提出的 MVENR 算法旨在用简单的矩阵融合思想学习出稳健的网络表示,使得学习得到的表示向量的性能与基于神经网络的网络表示学习的性能持平或者略优,而非提出一种性能远远优于基于神经网络的网络表示学习算法。在神经网络研究的浪潮中,验证使用传统的方法也能够达到优异的性能是本文研究的初衷。

## 2 问题定义

假设存在一个网络,该网络表示为  $G=(V,E)$ ,其中  $V$  为网络顶点集, $E$  为网络的边集,且  $E \in V \times V$ , $|V|$  为网络顶点集的大小,即顶点的个数。对于网络中的每个节点  $v \in V$ ,本文研究的目标是学习网络的表示向量  $r_v \in \mathbb{R}^k$ ,其中, $k$  为网络的表示向量的长度,其值远小于  $|V|$ 。另外, $r_v \in \mathbb{R}^k$  不仅能用于网络节点的分类任务,同时也可用于其他各类机器学习任务,如节点聚类、链路预测、推荐系统等。文中使用  $r_v$  表示网络表示学习的目标向量,并通过网络节点分类、可视化和案例分析等任务来验证其稳定性和可靠性。

文中所引入的 3 个网络特征视图的定义如下。

(1)连边权重视图(Link Weight View,LWV):元素值为已存在边的连接权重和未存在连边的节点对在未来产生连接的概率。

(2)网络结构视图(Network Structure View, NSV):DeepWalk 为分解矩阵  $M=(A+A^2)/2$ 。NSV 视图中,将矩阵  $M$  作为网络的结构特征视图。

(3)节点属性视图(Node Attribute View,NAV):由引文网络的节点文本构成,经过特征选择后形成属性特征视图。

## 3 算法理论

### 3.1 网络连边权重视图

链路预测是根据已知的网络节点和结构来预测两个未连边的节点之间在未来关联的可能性。这种预测包含对现有但未知连边的预测和未来连边的预测。复杂网络链路预测已经得到了物理和计算机科学界越来越多的关注。本文借助链路预测算法,学习网络中隐含的权重信息。本文将网络中存在边的连接权重和不存在边的未来连接概率引入到网络表示学习中,借此将节点之间的权重和概率融入到网络的表示中,使得权重和概率越大的节点对在表示空间中具有更近的距离。本文中使用的网络连边权重视图主要是通过链路预测算法所得,具体构建视图的流程如图 1 所示。

如图 1 所示,引文网络中,部分节点之间存在连边,部分节点之间不存在连边。节点对(1,2),(1,3)和(1,4)之间存在连边,且连边的权重值分别为 0.9,0.8 和 0.8。节点对(2,

3),(3,4)和(2,4)之间不存在连边,可以虚构节点之间的连边,即假设不存在连边的节点对在将来会产生连接关系。为了估算存在连边的节点之间的连接权重和不存在连边的节点之间的未来连接概率,文中引用了链路预测中的 MFI (Matrix Fo-rest Index)算法<sup>[20]</sup>。与其他链路预测算法相比,MFI 算法相对稳定,且在引文网络上的链路预测性能最优。

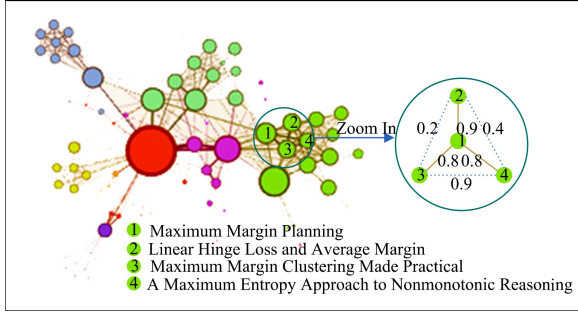


图 1 网络权重视图构建流程

Fig. 1 Construction process of network weight view

MFI 是基于矩阵森林理论被提出的,其定义如下:

$$S = (I + L)^{-1} \quad (1)$$

其中, $L$  为拉普拉斯矩阵, $I$  为单位矩阵。一般情况下,两个节点之间可能存在多条带权的连边。因此, $L$  可被定义为:

$$l_{xy} = \begin{cases} -\sum_p w_{xy}^p, & x \neq y \\ -\sum_{x \neq y} l_{xy}, & x = y \end{cases} \quad (2)$$

其中, $w_{xy}^p$  为节点  $v_x$  与节点  $v_y$  之间的第  $p$  条连边的权重。因此,节点  $v_x$  与节点  $v_y$  之间的相似度可通过如下定义计算:节点  $v_x$  和  $v_y$  属于同一棵以节点  $v_x$  为根节点的树的数目除以网络中所有仅含一个根节点的生成森林的数目。MFI 含参数的形式定义如下:

$$S = (I + aL)^{-1}, a > 0 \quad (3)$$

文中提出的网络权重视图是基于链路预测中的 MFI 算法计算所得,MFI 的计算公式如式(3)所示。仅通过网络的拉普拉斯矩阵,即可求得网络的权重视图矩阵。

### 3.2 网络结构视图

Levy 等<sup>[10]</sup>证明了基于 SGNS 模型的 Word2Vec 词向量表示学习算法的实质为矩阵分解 SPPMI 矩阵。SPPMI 矩阵的定义如下:

$$SPPMI_{i,j} = \log \frac{N(v_i, c_j) \cdot |D|}{N(v_i) \cdot N(c_j)} - \log n \quad (4)$$

其中, $n$  为负采样个数, $D$  表示语义空间中的词语总量; $N(v)$  为词语  $v$  在  $D$  中的频率, $N(c)$  为上下文  $c$  在  $D$  中的频率, $N(v, c)$  为  $(v, c)$  对在  $D$  中的频率,此处的频率为共现频率。

受 Word2Vec 矩阵分解算法的启发,Yang 等<sup>[11]</sup>从数学角度证明了 DeepWalk 算法的实质为对一个转移矩阵的复合式进行矩阵分解,即矩阵分解矩阵  $M$ 。 $M$  的表达式的定义如下:

$$M_{ij} = \log \frac{N(v_i, c_j)}{N(v_i)} \quad (5)$$

其中, $v_i$  定义为网络中的节点, $c_j$  定义为当前节点的上下文节点。DeepWalk 通过随机游走粒子在网络上的随机游走获取  $c_j$ 。给定网络  $G=(V, E)$ ,其中  $V$  为顶点集, $E$  为边集,定义  $D$  为随机游走粒子在网络上的随机游走产生的  $(v_i, c_j)$  对的集合。

DeepWalk 算法需要自定义随机游走的步长,假设步长为  $s$ ,则  $N(v_i)/|D|$  表示节点  $v_i$  在  $(v_i, c_j)$  对中出现的频率。根据 PageRank 的定义, $N(v_i)/|D|$  等于节点  $v_i$  的 PageRank 值。 $2sN(v_i, c_j)/N(v_i)$  为节点  $v_j$  在节点  $v_i$  周围  $s$  步之内出现的次数, $s$  表示随机游走步长。文中定义 PageRank 的转移矩阵为  $A$ , $A$  同时也是  $N(v_i)/|D|$  的存储矩阵。 $A$  的计算公式定义如下:

$$A_{ij} = \begin{cases} 1/d_i, & (i, j) \in E \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

其中, $d_i$  表示节点  $i$  的度。

另定义  $e_i$  为  $|V|$  维行向量,该行向量中,第  $i$  列元素为 1,其余全为 0。假设随机游走粒子将节点  $i$  作为初始节点,则  $e_i$  表示随机游走粒子的初始状态, $e_i A$  向量中的第  $j$  个值为游走粒子从节点  $i$  随机游走至节点  $j$  的概率值。同样, $e_i A^s$  表示随机游走粒子在  $s$  步内随机游走到目标节点  $j$  的概率值。综上, $[e_i (A + A^2 + A^3 + \dots + A^s)]_j$  表示节点  $v_j$  以节点  $v_i$  为中心的随机游走节点序列出现的次数,该随机游走的步长为  $s$ 。由此可得:

$$\frac{N(v_i, v_j)}{N(v_i)} = \frac{[e_i (A + A^2 + A^3 + \dots + A^s)]_j}{s} \quad (7)$$

因此:

$$M_{ij} = \log([e_i (A + A^2 + A^3 + \dots + A^s)]_j / s) \quad (8)$$

式(8)中计算  $M$  的开销较大,时间复杂度为  $O(|V|^3)$ 。

DeepWalk 在网络上的随机游走,获取网络上的  $(v_i, c_j)$  对集合,并将其输入到神经网络进行训练,从而有效地避免了直接计算矩阵  $M$ 。文中使用矩阵分解方法模拟 DeepWalk 算法,就不可避免地需要计算矩阵  $M$ 。Yang 等<sup>[13]</sup>已经证明了 DeepWalk 算法与矩阵分解式(8)等价,但是在衡量精度和速度要求后,提出了模拟 DeepWalk 算法仅需要分解矩阵  $M = (A + A^2)/2$  即可。如果网络是稠密的,甚至直接可以将矩阵  $M = A$  作为待分解的目标矩阵。因此,本文拟使用 Yang 等<sup>[13]</sup>的证明结果,将矩阵  $M = (A + A^2)/2$  作为网络的结构属性特征。分解矩阵  $M$  等同于采用 DeepWalk 算法获取网络的表示向量。

### 3.3 网络节点属性视图

网络节点属性是网络的重要组成部分。如果在网络特征挖掘中同时考虑网络的结构关系和网络节点的文本内容信息,则能更加准确地挖掘出反映网络真实情况的特征。

在科研引文网络中,节点的文本内容为论文的标题,TADW 算法<sup>[14]</sup>在网络表示学习中添加了节点的文本特征,其性能优于仅从结构学习得到的网络表示向量。受 TADW 算法的启发,本文尝试将网络节点的文本属性构建为网络的一个视图特征,然后将此视图与本文中的网络结构视图和网络连边权重视图进行融合。

网络节点属性视图的数据来源于网络节点的文本内容。对于单个数据集,构建网络节点属性视图的流程如下:

(1)首先统计该数据集中所有的词语,构建词典  $D$ ,词典格式为  $\langle \text{word}, id \rangle$ ,其中  $id$  满足  $id \in [0, |D|)$ 。

(2)对于每个节点的文本内容,以空格分割文本内容,返

回本节点文本包含的词语列表  $lwd$ 。

(3) 对于节点  $i$  的词语列表  $lwd$ , 从词典  $D$  中查询该词语的  $id$ , 构建  $id$  列表  $lid$ 。

(4) 对于节点  $i$ , 构建一个一行、 $|D|$  列的文本特征矩阵  $e_i$ 。

(5) 置  $e_i$  中的元素值,  $e_i(lid) = 1$ , 其余元素均为 0。

(6) 对于节点  $i \in [0, |D|)$ , 求得每个节点  $i$  对应的文本特征向量  $e_i$ 。

(7) 构建文本特征矩阵  $T$ ,  $T$  中的每个行向量为  $e_i$ 。

(8) 使用 SVDS 矩阵分解算法对  $T$  进行分解。  $T_{|V| \times |V|} \approx U_{|V| \times 200} \times S_{200 \times 200} \times (V_{200 \times |V|})^T$ , 文本特征向量的长度为 200 维。

(9) 将  $U_{|V| \times 200} \times S_{200 \times 200}^T$  矩阵视为本文中提出的网络节点属性视图。

以上内容为本文构建网络节点属性视图的具体流程。在步骤(8)中, 将矩阵降维到 200。因为对于不同的数据集, 词典  $D$  的大小不一致, 从而导致了矩阵的列维度不一致, 所以将不同数据集的网络节点属性视图列维度归化至统一的标准, 便于后续算法的性能对比。另外, 如果未对节点属性视图降维, 则该视图是一个非常稀疏的矩阵, 与其他视图进行融合时会引入大量的不相关特征。降维至 200 后, 保留了矩阵中重要的特征, 并去除了矩阵中的噪音, 有助于本文算法性能的

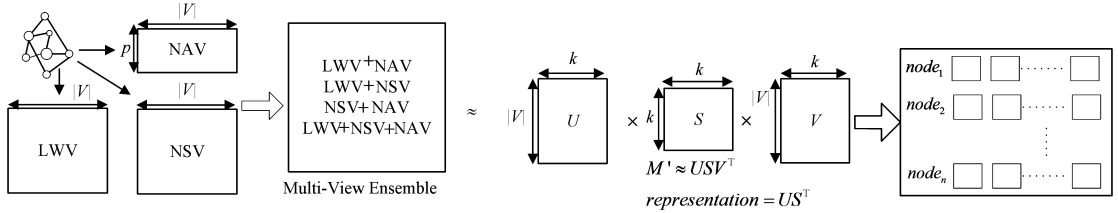


图 2 MVENR 算法框架图

Fig. 2 Framework of MVENR algorithm

关于矩阵分解, 本文中使用的是 SVDS 算法。相较于 SVD 算法, SVDS 算法具有以下优势: 1) SVDS 是 SVD 算法的变体, 虽然它们均是基于奇异值分解的方式分解矩阵, 但 SVDS 降低了计算复杂度; 2) SVDS 可以返回指定个数的最大特征值以及它们的特征行向量和列向量; 3) SVDS 具有更强的可定制性和可塑性。基于以上优点, 本文使用 SVDS 算法分解网络特征矩阵。

本文提供了 4 种视图集成方式:  $LWV + NAV$ ,  $LWV + NSV$ ,  $NSV + NAV$ ,  $LWV + NSV + NAV$ 。其中, “+”表示视图矩阵相乘, 在此处表示组合或集成之义, 并非矩阵求和。

如图 2 所示, 本文提出的 MVENR 算法的框架可简单分解为 4 个步骤, 具体算法的伪代码如算法 1 所示。

#### 算法 1 MVENR

Input: network edge list

Output: network representations

1. Get network edge list

2. Converted to adjacent matrix for edge list

3. View matrix generating

3.1. Calculate LWV by the contents of chapter 3.1

提升。文本特征矩阵的降维也可以被认为是对文本特征矩阵的去噪。

#### 3.4 MVENR 算法

本文提出的 MVENR 算法主要是融合网络的结构视图、节点文本视图和连边权重视图, 进而学习网络的表示向量。融合之后的网络结构特征既有了网络的结构特性, 又有了网络的节点文本特性和连边权重特性。DeepWalk 可以基于网络的结构学习网络的表示向量, 又可以通过将网络扩展为多层的异构网络来学习多特征融合的网络表示学习。上文中已证明基于深度学习的 DeepWalk 算法的实质为分解矩阵  $M = (A + A^2)/2$ 。采用 SVD 算法分解矩阵  $M = (A + A^2)/2$ , 获得的网络表示向量与基于 DeepWalk 算法获得的网络表示向量的性能相当。因此, 本文将矩阵  $M$  作为网络的结构视图, 使得网络的结构视图能够充分利用 DeepWalk 在网络表示学习中的优势。

对于不同视图的信息融合, 本文采用了矩阵相乘的方法。基于矩阵相乘的信息融合方法在推荐系统中已经取得了成功, 其推荐效果优于单个矩阵的推荐或者其他形式的矩阵融合。因此, 本文将推荐系统信息融合的方法引入到网络的表示学习算法之中。

文中提出的 MVENR 网络表示学习算法的具体框架如图 2 所示。

3.2. Calculate NSV by the contents of chapter 3.2

3.3. Calculate NAV by the contents of chapter 3.3

4. Choose a kind of view ensemble method from  $LWV + NAV$ ,  $LWV + NSV$ ,  $NSV + NAV$ ,  $LWV + NSV + NAV$  to construct feature matrix  $M'_{|V| \times |V|}$

5. Use SVD to factorize matrix  $M'_{|V| \times |V|}$

$$M'_{|V| \times |V|} \rightarrow U_{|V| \times k} \times S_{k \times k} \times V_{k \times |V|}$$

6. Gain the network representations

$$\text{representations} = U_{|V| \times k} \times S_{k \times k}^T$$

如算法 1 所示, 本文提出的 MVENR 算法的伪代码可细分为 6 个步骤。第 1 步和第 2 步是数据处理, 因此 MVENR 算法处理可简化为 4 个步骤。

步骤 1 使用 3.1 节中的算法提取网络的 LWV 视图特征矩阵; 使用 3.2 节中的算法提取网络的 NSV 视图特征矩阵; 使用 3.3 节中的算法提取网络的 NAV 视图特征矩阵。

步骤 2 选择视图集成方法, 获得信息集成后的矩阵  $M'$ 。

步骤 3 使用 SVDS 分解算法将目标矩阵  $M'_{|V| \times |V|}$  分解为  $U_{|V| \times k}$ ,  $S_{k \times k}$  和  $V_{k \times |V|}$  3 个矩阵相乘。

步骤 4 将矩阵  $U_{|V| \times k}$  和  $S_{k \times k}^T$  相乘, 得到网络中每个节

点的网络表示形式,即网络中每个节点的表示向量组成的矩阵为  $U_{|V| \times k} \times S_{k \times k}^T$ ,该矩阵是一个  $|V|$  行、 $k$  列的矩阵表示。

在步骤 3 中,  $|V|$  表示网络中节点的个数。 $k$  为特征值的个数,本文中可被认为是向量的长度。矩阵  $U$  是  $M'$  的奇异向量,  $S$  是一个对角矩阵,其中的元素为  $M'$  的奇异值,  $M'(M')^T$  的正交单位特征向量组成  $U$ , 特征值组成  $S^T S$ ,  $M'(M')^T$  的正交单位特征向量组成  $V$ , 特征值组成  $SS^T$ 。SVDS 被广泛应用于各类数据降维和推荐系统等任务中。

## 4 实验与评估

本文实验基于 3 个真实的引文网络数据集,使用网络节点分类、网络表示可视化、案例研究和链路预测等方式来评价本文所提 MVENR 算法的性能,并基于参数分析实验发现 MVENR 算法的最优参数组合。

### 4.1 数据集设置

本文实验基于 3 个真实的引文网络数据集,即 Citeseer, DBLP(V4) 和 Cora。本文从上述 3 个引文网络数据集网站下载原始数据,然后对其进行预处理操作,包括删除网络中孤立的顶点、建立节点之间的连接关系等。3 个真实网络数据集的统计指标如表 1 所列。

表 1 网络数据集的统计指标

Table 1 Statistical indicators of network datasets

数据集	节点数	边数	类别数	平均度	网络直径	平均路径长度	密度	平均聚类系数
Citeseer	3312	4732	6	2.857	8	2.02	0.001	0.080
DBLP(V4)	3119	39516	4	21.07	17	4.71	0.005	0.221
Cora	2708	5429	7	4.01	15	4.79	0.001	0.130

本文首先基于 MVENR 算法获得网络节点的网络,如表 1 所列。可以发现,Citeseer, DBLP(V4) 和 Cora 拥有数量几乎相同的网络节点,但网络数据集的边数不一致,Citesser 和 Cora 网络数据集的连边数量几乎相同,其边数量约为 5000,但 DBLP(V4) 数据集中连边数几乎是 Citesser 和 Cora 数据集的 6 倍,其边数量约为 4 万。另外,在拥有大致相同的节点数量的条件下,连边的数量越多,网络的密度越大,且网络的平均度也随之增大。例如,Citeseer 和 DBLP(V4) 的节点数约为 3000,但 DBLP(V4) 的连边数比 Citeseer 约多 6 倍,因此, DBLP(V4) 的网络密度是 Citeseer 的 5 倍, DBLP(V4) 的平均聚类稀疏是 Citeseer 的 3 倍左右。若连边数和节点数几乎一致,则网络的密度也几乎相同。虽然 DBLP(V4) 和 Cora 数据集的连边数的差别较大,但它们却拥有相同的网络直径和平均路径长度。由网络的平均度和密度可知,Citeseer 和 Cora 是一个稀疏网络,而 DBLP(V4) 是一个稠密网络。

### 4.2 对比算法

为了验证 MVENR 算法生成的网络表示向量具有稳定且高效的分类和预测能力,文中引入了现存的一些主流的网络表示学习算法和链路预测算法。若不特殊说明,本文中应用的对比算法的特征向量长度均设置为 200 维。相关对比算法的简介如下。

(1) DeepWalk。DeepWalk<sup>[5]</sup> 是当前应用得最多的网络表示学习算法,同时也是网络表示学习中的起源算法。基于网

络的结构特征,该算法使用了 Word2Vec 中的 Skip-Gram 模型和 Hierarchical Softmax 学习网络的特征表示。在该算法中,设置随机游走长度为 80,随机游走的数目为 10。

(2) LINE。LINE<sup>[22]</sup> 主要用来完成大规模网络表示学习的任务。类似于 DeepWalk, LINE 是一种局部特征获取算法,同时也应用了概率损失函数。LINE 中首次提出了一阶相似度与二阶相似度的概念,并提供了 1st LINE 和 2nd LINE 两种实现方式。本文使用 2nd LINE 学习和训练网络的表示向量。

(3) Node2vec。Node2vec<sup>[23]</sup> 用于解决网络中的结构等价和一般等价之间的平衡性。设置向量长度为 200,随机游走的数目为 10,随机游走的长度为 80。

(4) MFDW。MFDW 是 DeepWalk 算法的矩阵分解,其目标矩阵为  $M = (A + A^2)/2$ 。矩阵分解采用 SVDS 算法,该算法等同于本文中提出的基于 NSV 视图的网络表示学习。

(5) MMDW。MMDW 同样是基于 DeepWalk 矩阵分解形式的一种算法。该算法针对矩阵分解的目标矩阵为  $M = (A + A^2)/2$ 。为了使得学习得到的网络表示向量具有更好的分类能力,MMDW 算法使用最大间隔优化学习得到  $W$  矩阵。因此,将 SVM 中的最大间隔理论和网络表示学习理论进行了有效融合。

(6) TEXT。本文中网络节点的文本特征向量定义为  $Y \in \mathbb{R}^{|V| \times 200}$ , 该向量的长度为 200。该方法是一种单纯的基于文本特征的网络表示学习算法,等同于本文中提到的基于 NAV 视图的网络表示学习。

(7) TADW。TADW 同样是基于矩阵分解的网络表示学习算法,其分解的目标矩阵为  $M = (A + A^2)/2$ 。类似于 MMDW 算法,TADW 在矩阵分解的过程中加入了节点的文本特征来辅助分解目标矩阵,使得分解得到的网络表示向量中既含有网络的结构信息,又含有节点的文本属性。

(8) MF-MFI。该算法是一种基于 MFI 相似度矩阵分解的算法,等同于本文中提到的基于 LWV 视图的网络表示学习。

### 4.3 实验结果与分析

首先训练网络的表示向量,然后基于训练所得的表示向量学习一个 SVM 分类器,并使用不同的训练率评估网络表示向量的性能。训练数据的训练率的变化范围为 10% ~ 90%。对于训练数据,随机且不重复地从整个数据集上获取数据作为训练集,直到训练集数量达到设定的训练率(训练集占总数据集的比例),并将其余的数据作为测试集。本文中,网络节点分类准确率均采用 10 次重复计算求平均值的方式获得。网络节点的表示向量长度和文本特征向量长度均设置为 200。具体的结果如表 2—表 4 所列。

从表 2—表 4 可以发现,网络的密度或平均度较小时,如果从结构视图挖掘网络的特征,该特征会因为网络稀疏得不到充分的训练和挖掘,从而导致基于网络结构的网络表示算法的性能较差。反之,如果网络密度或平均度较大,则基于结构的网络特征挖掘算法能够较充分地从网络结构信息中挖掘到特征,此时,基于结构视图的网络表示学习算法的性能较

好。因此,本文集成算法的性能会受限于结构视图中获取的特征质量,但网络连边权重视图和节点属性视图的参与在很

大程度上弥补了结构视图的稀疏性,因而在稀疏网络中可得到稳健且具有较强分类能力的网络表示向量。

表2 Citeseer数据集上节点的分类准确率  
Table 2 Classification accuracy of vertex on Citeseer

(单位: %)

LABELED NODES	10	20	30	40	50	60	70	80	90
DeepWalk	48.31	50.36	51.33	52.31	52.85	53.33	52.98	53.47	53.71
LINE	39.82	46.83	49.02	50.65	53.77	54.20	53.87	54.67	53.82
MMDW	55.49	60.70	63.66	65.27	66.02	69.14	69.34	69.47	69.72
TADW	68.19	70.03	71.67	72.45	73.76	74.06	74.48	74.74	75.59
Node2vec	54.38	57.29	58.64	59.53	59.63	59.88	60.43	61.36	62.42
TEXT(NAV)	57.76	66.02	69.77	71.29	72.14	72.90	72.68	72.34	73.83
MFDW(NSV)	49.78	54.80	56.66	56.75	57.90	58.32	58.60	58.29	57.07
MF-MFI (LWV)	52.98	57.95	59.06	60.91	61.28	62.04	62.53	62.58	64.29
MVENR (LWV+NAV)	64.89	71.11	73.33	75.49	76.54	76.38	76.81	77.05	76.46
MVENR (LWV+NSV)	52.73	58.45	59.44	60.64	61.42	62.06	62.84	63.42	63.08
MVENR (NSV+NAV)	63.70	70.40	72.18	73.48	74.81	75.22	75.33	75.83	76.49
MVENR (LWV+NSV+NAV)	64.19	69.59	72.24	72.99	74.26	74.32	74.93	75.43	74.65

表3 DBLP(V4)数据集上节点的分类准确率  
Table 3 Classification accuracy of vertex on DBLP(V4)

(单位: %)

LABELED NODES	10	20	30	40	50	60	70	80	90
DeepWalk	81.84	82.41	83.25	83.74	83.98	84.24	84.55	84.26	83.53
LINE	79.13	79.81	80.41	81.22	82.95	83.39	83.04	84.74	83.85
MMDW	79.70	82.05	84.23	84.84	83.45	85.42	84.96	85.78	84.49
TADW	81.09	82.43	83.42	83.74	84.16	84.40	84.91	85.26	86.05
Node2vec	82.71	83.66	84.07	84.51	84.18	84.71	85.28	84.99	84.69
TEXT(NAV)	60.69	68.15	70.62	72.76	73.79	74.73	75.37	75.16	74.85
MFDW(NSV)	75.06	80.82	83.00	83.96	84.70	84.94	85.72	84.62	85.11
MF-MFI (LWV)	71.88	78.24	79.90	80.48	81.34	82.29	82.19	83.16	83.02
MVENR (LWV+NAV)	75.46	81.67	84.53	84.98	84.77	86.34	85.82	86.00	86.72
MVENR (LWV+NSV)	75.71	80.54	82.65	83.81	84.32	84.68	84.63	85.74	84.41
MVENR (NSV+NAV)	74.72	81.70	83.70	84.33	84.50	85.02	84.90	85.55	86.43
MVENR (LWV+NSV+NAV)	75.76	82.11	83.72	84.68	85.22	85.62	85.76	85.44	86.36

表4 Cora数据集上节点的分类准确率  
Table 4 Classification accuracy of vertex on Cora

(单位: %)

LABELED NODES	10	20	30	40	50	60	70	80	90
DeepWalk	73.29	75.46	76.19	77.49	77.89	77.83	78.86	79.05	78.62
LINE	65.13	70.17	72.20	72.92	73.45	75.67	75.25	76.78	79.34
MMDW	73.61	79.99	80.43	81.92	83.76	84.97	86.39	86.70	87.45
TADW	80.69	81.70	84.47	85.94	86.35	86.27	85.97	87.36	87.70
Node2vec	76.30	79.26	80.43	80.70	81.13	81.26	82.18	81.63	82.81
TEXT(NAV)	57.70	67.26	70.94	73.05	74.47	74.74	77.04	76.17	77.00
MFDW(NSV)	66.38	75.52	78.78	80.54	82.09	81.93	82.62	81.57	83.81
MF-MFI (LWV)	72.25	78.27	81.25	81.73	82.74	83.49	84.64	84.39	85.55
MVENR (LWV+NAV)	72.61	82.40	85.20	86.52	87.33	87.62	88.36	88.26	88.11
MVENR (LWV+NSV)	71.19	78.81	80.71	82.69	83.30	83.50	84.18	84.38	85.29
MVENR (NSV+NAV)	71.58	82.10	84.41	85.44	86.35	87.32	87.38	87.28	87.55
MVENR (LWV+NSV+NAV)	73.80	82.57	84.73	86.02	86.61	87.46	87.51	88.17	87.78

基于上述实验,我们还发现了以下结论:

(1)LWV+NAV 的组性能优于其他组合形式。LWV+NSV+NAV 3种视图的组合效果不佳,差于LWV+NAV视图的组合效果,但是优于LWV+NSV和NSV+NAV的组合。另外,多视图集成的方法优于本文中的其他对比算法,由此可得,基于多视图集成的网络表示能够产生稳健且高效的网络嵌入。

(2)LWV+NSV视图的组合效果在3类数据集上都表现出了最差的性能。即,不论NSV的性能是好还是差,在3个

数据集上,LWV+NSV的性能都是最差的。但是NSV与NAV视图组合时,不论NAV视图的性能是好还是差,NSV+NAV视图的组合性能总是好于LWV+NSV。

(3)NSV+NAV是一种网络结构和文本属性相结合的方法,TADW同样也是一种结构和文本属性相结合的方法。但是在矩阵分解时,它们所采用的策略不一样:NSV+NAV采用的是先组合后分解的策略,TADW采用的是边分解边组合的策略。结果表明,NSV+NAV的性能优于TADW的性能,即先组合后分解的策略优于边分解边组合的策略。

(4)将几个性能不佳的网络特征矩阵进行组合,分解后得到的网络表示优于仅对一个视图进行分解得到的表示,也优于基于随机游走得到的网络表示或者其矩阵分解表示形式。该思路与集成学习中几个弱分类器的组合学习可得到一个强分类器的假设是一致的。

(5)LWV 是一种从结构获得的矩阵森林指标,其实质为一种基于结构的指标,因此,与基于结构的 MSV 特征视图集成时,集成的性能劣于其他集成方式。从而可以发现,多视图集成学习时,不同的视图应该遵循结构的独立性和特征的差异化,尽量从不同的角度收集网络的特征,避免两种特征来源相同的视图进行集成学习。

#### 4.4 参数分析

本文实验中,向量长度主要涉及到文本特征的降维长度和最终的网络表示向量的长度。首先,设置训练集的学习率为 0.5,即一半数据为训练集,另一半数据为测试集。实验指标为衡量网络节点分类的准确率。文本向量长度的取值为  $[25, 50, 100, 200, 300, 500]$ ,最终集成后得到的网络节点的表示向量长度取值为  $[25, 50, 100, 200, 300, 500]$ 。分别设置了不同的文本特征向量长度和最终的网络表示向量长度,然后进行节点分类实验。每次实验均重复进行 10 次再求平均值。通过该实验可以观察到文本特征向量长度与网络表示向量长度之间的关系,具体如图 3 所示。

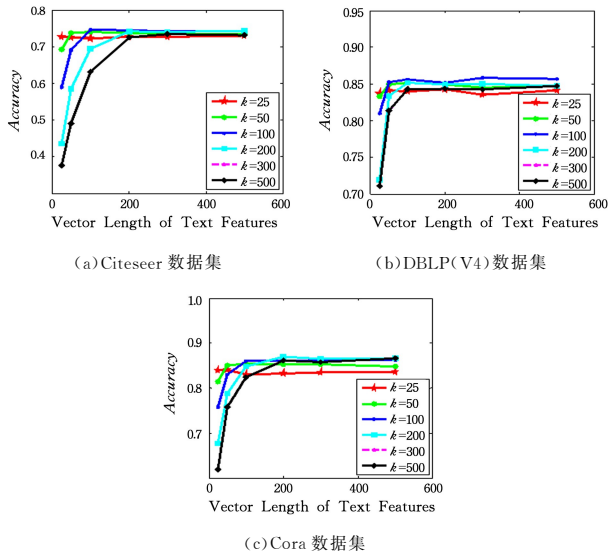


图 3 向量长度的影响分析

Fig. 3 Influence analysis of representation length

如图 3 所示,图中  $k$  为网络表示向量的长度。网络节点分类的准确率随着文本特征长度的增长而增长,但是到了一定的阈值后,准确率趋于稳定。在 Citeseer 数据集上,当文本特征向量长度为 200 时,准确率趋于稳定;在 DBLP(V4) 和 Cora 数据集上,文本特征向量长度为 100 时,准确率趋于稳定。产生该现象的主要原因是, Citeseer 是一个稀疏的网络,其网络平均度低于 DBLP(V4) 和 Cora 数据集。

另外,文中提出了 4 种视图集成方法,每种视图的集成都取得了较好的性能,超过了本文中列举的对比算法。为了研

究不同视图集成方式与文本特征向量长度和集成向量长度之间的关系,本节对视图集成受集成后向量长度的影响与变化进行了可视化,同时也可可视化了视图集成受文本特征向量长度的影响与变化。具体结果如图 4 和图 5 所示。

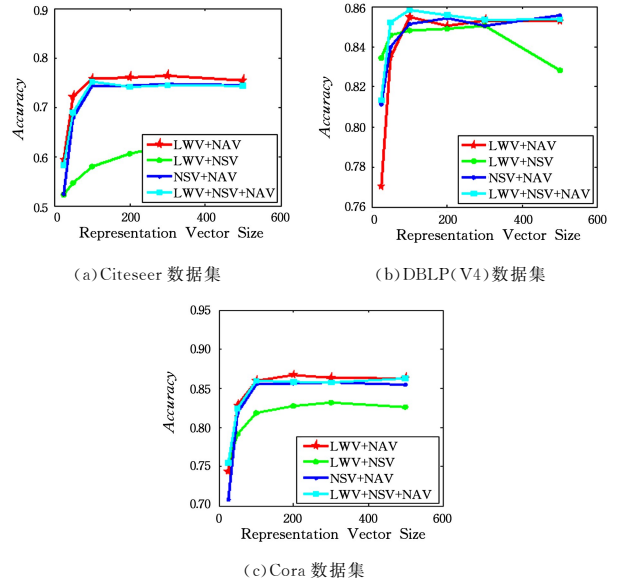


图 4 表示向量长度与网络节点分类准确率之间的关系

Fig. 4 Relationship between representation length and classification accuracy

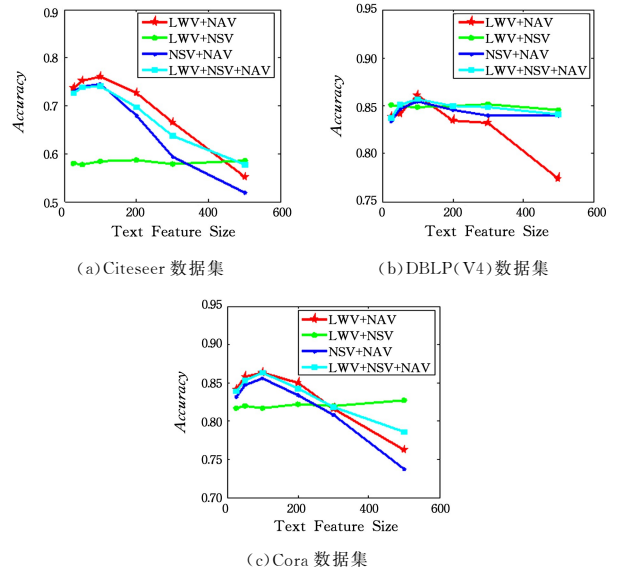


图 5 文本特征向量长度与网络节点分类准确率之间的关系

Fig. 5 Relationship between text vector length and classification accuracy

图 4 表示出网络表示向量长度与网络节点分类准确率之间的关系,图 5 表示出文本特征向量长度与网络节点分类准确率之间的关系。可以发现,网络表示向量长度和文本特征向量长度对网络节点分类的准确率影响不同。固定文本特征向量长度为 100,当网络表示向量长度小于 100 时,网络节点分类的准确率呈增长趋势;当网络表示向量长度大于 100 时,除 LWV+NSV 集成方式外,其他集成方式的网络节点分类准确率呈现降低趋势,但 LWV+NSV 集成的网络表示节点

分类准确率却有缓慢的增长趋势。因此可得,网络节点分类的准确率在一定范围内呈快速上升趋势,到达一定阈值后,增长速度趋于平缓,或呈现降低趋势。出现这种现象的原因是,Citeseer和Cora网络的平均度较小,所以下降速度较快,而DBLP(V4)网络的平均度较高,所以下降速度较慢。LWV+NSV视图的集成方法随着文本特征向量长度的变化呈现出缓和的上升或者下降趋势,准确率变化幅度较小。出现该现象的主要原因是,在LWV+NSV视图的集成方式中未有文本特征的参与,因此文本特征的长度不影响分类性能。LWV+NSV集成方式的准确率变化主要来源于矩阵分解过程中随机因子造成的波动。对于本文采用的SVM分类算法,并非输入的特征向量越长,准确率就越高。

#### 4.5 集成向量可视化

本文从DBLP(V4)数据集中随机选取4个类别,每个类别随机选取150个节点。然后使用T-SNE<sup>[21]</sup>可视化算法将每个数据集中的600个节点投影到二维平面,不同类别的网络节点用不同灰度表示。为了说明不同视图集成方式产生的网络表示向量的聚类能力,图6展示出将LWV+NAV, LWV+NSV, NSV+NAV和LWV+NSV+NAV视图集成算法训练的600个节点的网络表示向量投影到二维平面的可视化结果。从图6可知,在网络节点表示向量的二维可视化中,稠密的DBLP(V4)网络能够产生较好的网络表示特征,产生的表示向量能够在聚类任务中表现出较好的性能,能够准确地发现不同类别之间的边界。另外,网络节点表示向量的质量决定着节点分类和聚类的能力,LWV+NSV和LWV+NSV+NAV视图集成方式在网络节点分类任务中均表现出了很好的性能,同样地,它们在二维可视化任务中也表现出了

较好的聚类能力。由此可知,稳健、高效的网络特征表示在多种任务中均能够产生优异的性能。

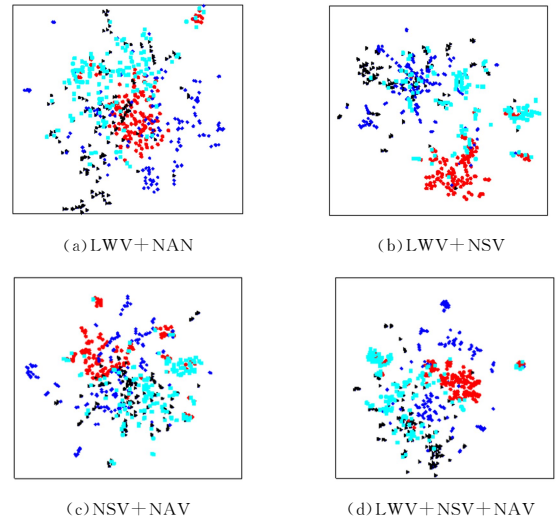


图6 DBLP(V4)数据集上视图集成方式对表示向量影响的可视化  
Fig. 6 Influence analysis of representation vector using various view ensemble methods on DBLP(V4)

#### 4.6 案例研究

为了验证本文提出的MVENR算法能够产生稳健、高效的网络表示向量,充分反映节点之间的关联关系,本文在DBLP(V4)数据集中随机选取一个目标节点,该节点的标题为“Factorial Hidden Markov Models”。然后通过计算余弦相似度值,得到与该标题节点相似度值最高的5个邻居节点,并获取这5个节点的标题。在本节实验中,设置网络节点表示的长度为200,训练率为0.9。实验结果如表5所列。

表5 最相关邻居节点标题的案例研究

Table 5 Case study on nearest vertex

Title	Sim.	label
DeepWalk		
Learning for Control from Multiple Demonstrations	0.9367	人工智能领域
Robot Learning from Demonstration	0.8953	人工智能领域
Apprenticeship Learning via Inverse Reinforcement Learning	0.8895	人工智能领域
Dynamic Preferences in Multi-criteria Reinforcement Learning	0.8653	人工智能领域
Algorithms for Inverse Reinforcement Learning	0.8152	人工智能领域
MVENR		
Relational Markov Models and Their Application to Adaptive Web Navigation	0.8015	人工智能领域
Learning Associative Markov Networks	0.7832	人工智能领域
Self-Similar Layered Hidden Markov Models	0.7359	人工智能领域
Rk-hist: An R-tree Based Histogram for Multi-Dimensional Selectivity Estimation	0.7262	人工智能领域
Maximum Entropy Markov Models for Information Extraction and Segmentation	0.7243	人工智能领域

如表5所列,本文获得了与标题的节点最相关的5个节点,并给出了相似度。DeepWalk算法和MVENR算法都能准确地识别节点的类别标号,但是它们得到的最相关的节点排序却不同。DeepWalk算法从网络的结构视角挖掘节点之间的关联关系,因此更多地考虑了节点之间的引用关系。本文提出的MVENR算法在基于结构的网络表示学习中融入了节点之间存在的连边权重、不存在边的未来连接概率和节点的文本特征,因此训练所得的网络表示会倾向于突出这3个属性之间的相似度。例如,在本文算法所计算的前5个最

相关节点的标题中有4个含有“Markov”,因此与目标节点的标题具有文本相似属性;其中有一个节点的标题中未包含“Markov”,根据MVENR算法可知,目标节点与该节点之间具有较大的连边权重或未来连接概率,或这两个节点之间具有较多的共同邻居,能够在随机游走过程中产生较近的关联。

**结束语** 文中提出了一种基于多视图集成的网络表示学习算法;为了评估单个视图对网络表示向量的贡献,又提出了LWV+NAV, LWV+NSV, NSV+NAV, LWV+NSV+NAV 4种不同的视图集成方式。实验结果表明,LWV+

NAV 视图集成的网络表示总能获得最好的性能, LWV + NSV + NAV 视图集成的网络表示学习算法优于 NSV + NAV 集成算法, LWV + NSV 视图集成的网络表示算法的性能劣于以上 3 种视图集成方式。另外, 本文基于网络节点分类实验还发现了一些有趣的结论: 从网络的不同角度获取网络的特征构建视图, 之后将不同的视图进行特征融合, 其性能会优于基于单个视图的网络表示学习算法。另外, 本文的实验结果充分表明了, 基于简单而高效的信息矩阵融合和分解理论能够产生稳健且具有判别力的网络表示向量, 其性能与部分基于神经网络的网络表示学习算法的性能持平。如果将多个视图进行信息融合, 再使用矩阵分解理论学习网络的表示向量, 其性能会优于部分使用深度学习框架的基于网络结构的表示学习算法。本文中的网络节点可视化、案例研究等实验也充分表明了, 本文提出的网络表示学习算法在各类机器学习任务中都具有较强的竞争力。

### 参 考 文 献

- [1] TSOUKAS G, KATAKIS I, TANIAR D. Multi-label classification: an overview [J]. *International Journal of Data Warehousing & Mining*, 2007, 3(3): 1-13.
- [2] LIBENNOWELL D, KLEINBERY J. The link-prediction problem for social networks [J]. *Journal of the Association for Information Science and Technology*, 2007, 58(7): 1019-1031.
- [3] ZHAO W X, HUANG J, WEN J R. Learning distributed representations for recommender systems with a network Embedding approach [C] // *Asia Information Retrieval Symposium*. Berlin: Springer, 2016: 224-236.
- [4] YU X, REN X, SUN Y, et al. Personalized entity recommendation: a heterogeneous information network approach [C] // *ACM International Conference on Web Search and Data Mining*. NY: ACM, 2014: 283-292.
- [5] PEROZZI B, AI-RFOU R, SKIENA S. DeepWalk: Online Learning of Social Representations [C] // *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. NY: ACM, 2014: 701-710.
- [6] PAN S, WU J, ZHU X, et al. Tri-party deep network representation [C] // *International Joint Conference on Artificial Intelligence*. NY: ACM, 2016: 1895-1901.
- [7] LI Q, LIU D. Research of music recommendation system based on user behavior analysis and word2vec user emotion extraction [C] // *International Conference on Intelligent and Interactive Systems and Applications*. Berlin: Springer, 2017: 469-475.
- [8] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space [C] // *International Conference on Learning Representations*. Palo Alto: AAAI, 2013: 1-12.
- [9] CHEN Y, PEROZZI B, SKIENA S. Vector-based similarity measurements for historical figures [M] // *Similarity Search and Applications*. Berlin: Springer, 2015.
- [10] LEVY O, GOLDBERY Y. Neural word embedding as implicit matrix factorization [M] // *Advances in Neural Information Processing Systems*. Massachusetts: MIT Press, 2014: 2177-2185.
- [11] YANG C, LIU Z. Comprehend deepwalk as matrix factorization [R/OL]. (2015-01-02) [2018-01-15]. <http://pdfs.semanticscholar.org/0edb/7368b6f14d1b4e3a062cb2fe96e9ae50e111.pdf>.
- [12] TU C, ZHANG W, LIU Z, et al. Max-margin deepwalk: discriminative learning of network representation [C] // *International Joint Conference on Artificial Intelligence*. Palo Alto: AAAI, 2016: 3889-3895.
- [13] YANG C, LIU Z Y, ZHAO D L, et al. Network representation learning with rich text information [C] // *International Conference on Artificial Intelligence*. Palo Alto: AAAI, 2015: 2111-2117.
- [14] NATARAJAN N, DHILLON I S. Inductive matrix completion for predicting gene-disease associations [J]. *Bioinformatics*, 2014, 30(12): 60-68.
- [15] KIPF T N, WELING M. Semi-supervised classification with graph convolutional networks [R/OL]. (2016-09-09) [2017-12-10]. <http://www.ics.uci.edu/~welling/publications/papers/Submitted2016-SSL-GCNN.pdf>.
- [16] LI J, ZHU J, ZHANG B. Discriminative deep random walk for network classification [C] // *Annual Meeting of the Association for Computational Linguistics*. NY: ACL, 2016: 1004-1013.
- [17] YANG Z L, COHEN W, SALAKHUTDINOV R. Revisiting semi-supervised learning with graph embeddings [C] // *International Conference on Machine Learning*. NY: ACM, 2016: 40-48.
- [18] WANG X, CUI P, WANG J, et al. Community preserving network embedding [R/OL]. (2017-02-04) [2017-11-24]. <http://www.aaai.org/ocs/index.php/AAAI/AAAI17/paper/download/14589/13763>.
- [19] YANG C, SUN M S, LIU Z Y, et al. Fast network embedding enhancement via high order proximity approximation [C] // *International Joint Conference on Artificial Intelligence*. AAAI Press, 2017: 3894-3900.
- [20] CHEBOTAREY P, SHAMIS E. The matrix-forest theorem and measuring relations in small social groups [J]. *Automation & Remote Control*, 2006, 58(9): 1505-1514.
- [21] MAATEN L V D, HINTON G. Visualizing Data using t-SNE [J]. *Journal of Machine Learning Research*, 2008, 9: 2579-2605.
- [22] TANG J, QU M, WANG M, et al. LINE: Large-scale information network embedding [R/OL]. (2013-03-12) [2017-12-12]. <https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/frp0228-Tang.pdf>.
- [23] GROVER A, LESKOVEC J. node2vec: Scalable Feature Learning for Networks [C] // *the 22nd ACM SIGKDD International Conference*. NY: ACM, 2016: 855-864.