

基于邻节点和关系模型优化的网络表示学习

冶忠林^{1,2,3} 赵海兴^{1,2,3} 张科^{1,3} 朱宇^{1,3} 肖玉芝^{1,3}

¹(青海师范大学计算机学院 西宁 810008)
²(陕西师范大学计算机科学学院 西安 710119)
³(藏文信息处理教育部重点实验室(青海师范大学) 西宁 810008)
(zhonglin_ye@foxmail.com)

Network Representation Learning Using the Optimizations of Neighboring Vertices and Relation Model

Ye Zhonglin^{1,2,3}, Zhao Haixing^{1,2,3}, Zhang Ke^{1,3}, Zhu Yu^{1,3}, and Xiao Yuzhi^{1,3}

¹(College of Computer, Qinghai Normal University, Xining 810008)
²(College of Computer Science, Shaanxi Normal University, Xi'an 710119)
³(Key Laboratory of Tibetan Information Processing (Qinghai Normal University), Ministry of Education, Xining 810008)

Abstract Network representation learning aims at embedding the network topology structures, vertex contents and other information of networks into the low-dimensional vector space, which thus provides an effective tool for network data mining, link prediction and recommendation system etc. However, the existing learning algorithms based on neural networks neglect the location information of the context vertices. Meanwhile, this kind of algorithms ignore the semantic associations between vertices and texts. Therefore, this paper proposes a novel network representation learning algorithm using the optimizations of neighboring vertices and relation model (NRNR). NRNR first uses the neighboring vertices to optimize the learning procedure, consequently, the location information of the vertices in the context windows is embedded into the network representations. In addition, NRNR first introduces the relational modeling from knowledge representation learning to learn the structure features of the networks, and the text contents between vertices are thus embedded into the network representations with the form of relational constraints. Moreover, NRNR proposes a feasible and effective network representation joint learning framework, which integrates the above two goals into a unified optimization objective function. The experimental results show that the proposed NRNR algorithm is superior to all kinds of baseline algorithms applied to the network node classification tasks in this paper. In network visualization tasks, the network representations obtained by NRNR algorithm show a distinct clustering boundary.

收稿日期:2018-08-14;修回日期:2019-06-05
基金项目:国家自然科学基金项目(11661069, 61763041, 11801296);长江学者和创新研究团队项目(IRT_15R40);青海省自然科学基金项目(2017-ZJ-949Q);中央高校基本科研业务费专项资金项目(2017TS045)
This work was supported by the National Natural Science Foundation of China (11661069, 61763041, 11801296), the Program for Changjiang Scholars and Innovative Research Team in Universities (IRT_15R40), the Natural Science Foundation of Qinghai Province of China (2017-ZJ-949Q), and the Fundamental Research Funds for the Central Universities (2017TS045).
通信作者:赵海兴(h.x.zhao@163.com)

Key words network representation learning; network embedding; network representation; vertex vector; network feature learning

摘 要 网络表示学习旨在将网络的拓扑结构、节点内容和其他信息嵌入到低维度的向量空间中,从而为网络数据挖掘、链路预测和推荐系统提供一种有效的工具.然而,现有的基于神经网络的表示学习算法即忽略了上下文节点的位置信息,又忽略了节点与文本之间的语义关联.因此,基于以上 2 点,提出了一种新颖的基于邻节点和关系模型优化的网络表示学习算法(network representation learning algorithm using the optimizations of neighboring vertices and relation model, NRNR).首先,该算法首次采用当前节点的邻居节点优化网络表示学习模型,使得上下文窗口中节点的位置信息被嵌入到网络表示中;其次,该算法首次引入知识表示学习中的关系模型建模节点之间的结构特征,使得节点之间的文本内容以关系约束的形式嵌入到网络表示中;再次, NRNR 提出了一种可行且有效的网络表示联合学习框架,将上述 2 种目标融入到一个统一的优化目标函数中.实验结果表明: NRNR 算法在网络节点分类任务中优于各类对比算法,在网络可视化中, NRNR 算法学习得到的网络表示展现出了明显的聚类边界.

关键词 网络表示学习;网络嵌入;网络表示;节点向量;网络特征学习

中图法分类号 TP182

随着计算机处理性能的提升和网络数据规模的不断增长,对网络数据挖掘的重视程度也越来越高.网络数据挖掘不仅可以挖掘出隐藏在数据之下的真实价值,更可以推动改变现实生活的进程.网络数据挖掘主要面向对象为各类现实生活中的数据网络,例如:社交网络、引文网络、博客网络和通信网络等.这些网络统称为信息网络,且具有规模大、数据杂、噪音多等特点^[1].目前,信息网络的数据挖掘是研究的热点和前沿,而深度学习、终生学习、强化学习等技术的提出更是将信息网络的数据挖掘引入了更深层次的研究范畴.网络数据挖掘具有重要的意义.例如在社交网络中,通过挖掘好友出行记录和文本内容可以预测游客未来出行安排,该预测结果可帮助旅游公司更加精准地推荐旅游服务信息.而信息网络与自然语言处理结合进行产品和服务的推荐是目前挖掘数据挖掘最有价值的应用.单纯地使用网络结构数据或者使用文本信息进行数据挖掘,则难以完全反映出真实网络的特征.因此,将网络结构特征和文本特征进行联合的学习和预测是数据挖掘致力于解决的难题.

信息网络数据挖掘首先需要面对的难题是如何恰当地表示网络数据.如果仅考虑网络的结构特征,则可以使用网络的邻接矩阵.如果仅考虑网络节点的文本特征,则可以将文本转化为共现矩阵.如果同时考虑网络的结构特征和网络节点的文本特征,则可以将 2 个矩阵进行列拼接或者降维后再拼接.这

种简单的矩阵拼接虽然包含了 2 种不同数据的特征,但是却没有充分地考虑 2 种数据之间的关系,即关系被忽略.不论何种处理方式,矩阵的运算会导致模型的训练产生较高的计算复杂度,另外,在大规模的网络数据挖掘任务中也不具有可行性.

网络表示学习算法对上述任务提出了可行的解决方案.网络表示学习通过神经网络对网络节点之间的结构进行编码,能够处理大规模网络结构特征编码任务.其中最经典的算法为 DeepWalk^[2] 算法,之后基于 DeepWalk 算法和思路提出了各种改进算法.然而现有的基于神经网络的表示学习算法将当前节点的所有上下文节点平等对待,使得在上下文窗口中的节点对当前中心节点具有同等地位的影响,即忽略了上下文节点的位置信息.例如在神经网络模型中当前窗口大小设置为 5 时,则当前节点前后的各 2 个节点作为其上下文节点.在神经网络训练过程中,上下文节点对 (n_{-2}, n_0) , (n_{-1}, n_0) , (n_0, n_1) , (n_0, n_2) 被输入到神经网络模型中通过节点对共现不断地调整其向量中的元素值.该过程使得常出现于节点对中的节点在向量表示空间中拥有更近的空间表示距离,未出现于节点对中的节点在向量表示空间中具有更远的空间表示距离.但是,该训练过程并未考虑 n_{-2}, n_{-1}, n_1, n_2 与 n_0 之间的距离.此时模型采用的是上下文结构无关性假设,采用这个假设的目的是为了加速模型训练速度.在文献[3-4]中已经验证了当窗口大小设置为 5 时,上下文对当

前元素的影响达到最大.过大的窗口和过小的窗口都会影响模型的泛化和学习能力.因此,本文在采用神经网络模型训练网络表示的过程中,首次对网络采样的上下文节点对进行深入的优化,即在上下文窗口内对当前节点的邻居节点进行优化.本文同样设置神经网络采样上下文节点的窗口大小为 5,此时 n_0 的上下文节点为 n_{-2}, n_{-1}, n_1 和 n_2, n_{-1}, n_1 与 n_0 之间的距离大于 n_{-2} 和 n_2 与 n_0 之间的距离.因此,训练所得的网络表示中, n_{-1} 和 n_1 的网络表示比 n_{-2} 和 n_2 的网络表示在向量空间中与 n_0 的网络表示具有更近的空间距离.为了实现该目标,本文优化了当前节点的 1 阶邻节点,即优化了 n_{-1}, n_1 对 n_0 的学习过程.2 阶邻居节点 n_{-2}, n_2 并未做任何处理.因为,优化了 n_{-1}, n_1 与 n_0 之间的学习强度,则同时自然地降低了 n_{-2}, n_2 与 n_0 之间学习强度.对于 n_0 ,最终的结果是 n_{-1} 和 n_1 被训练出了更近的位置关系, n_{-2} 和 n_2 被训练出了较远的位置关系.本文对 1 阶邻节点所采用的优化过程不同于 node2Vec^[5] 和 Walklets^[6],这 2 类算法采用通过改变随机游走策略的方法改变神经网络的输入.而本文并未改变神经网络的输入,而是在神经网络训练的过程中优化随机游走序列中 1 阶邻节点对当前节点的贡献.

另外,现有的网络表示学习算法通常将节点文本作为特殊的节点或使用另外一个神经网络训练文本向量.例如 TADW^[7] (text-associated deep walk) 将网络节点的文本内容转化为词共现矩阵,之后使用降维方法得到文本的低维度特征.为了将文本特征和网络结构特征有效地融合, TADW 引入了诱导矩阵补全算法^[8],该算法将网络结构特征作为待分解的目标特征矩阵,同时将文本特征作为目标矩阵的辅助分解矩阵. TriDNR^[9] 使用第 1 个 3 层神经网络建模节点与节点之间的关系,并使用第 2 个 3 层的神经网络建模节点与其文本中的词语之间的关系. TriDNR 同样使用第 2 个 3 层神经网络建模标签类别与文本中词语之间的关系. CENE^[10] 仍然采用 DeepWalk^[2,11-12] 建模节点与节点之间的关系,并使用循环神经网络 (recurrent neural network, RNN) 和双向循环神经网络 (bidirectional recurrent neural network, BiRNN) 建模节点与文本内容之间的关系. TADW 和 CENE 方法在构建文本特征时均采用了独立于建模节点关系之外的策略, TriDNR 采用了将文本中的词语作为特殊的网络节点的策略.这 3 类方法在网络表示学习过程中采用了不同的策略建模节点与节点、节点与文本内容之间的关系.因此,

本文在以上网络表示学习融入文本特征的研究基础上,引入了知识表示学习中的关系模型.利用关系模型可以在建模节点与节点关系的同时考虑节点文本之间的关系类型.为了实现该目标,本文引入了节点三元组的概念.而知识三元组 (subject, predict, object)^[13] 常被用来表示知识及其关系,并在知识库中被大量应用.在关系模型中,知识被定义为关系三元组 (h, r, t) ^[14] 的形式.本文采用关系三元组 (h, r, t) 定义节点与节点关于文本的关系. h 项和 t 项表示头节点和尾节点, r 表示节点与节点之间在文本上拥有的共现词语,即 2 个节点的文本特征中含有共同的词语则表示 2 个节点之间具有了文本关联性.本文并不采用单独的关系模型建模节点与文本之间的关系,而是采用统一的目标函数将节点关系建模与节点与文本关系建模融合在一起进行训练,这样处理既可以保留节点关系建模过程,又可以引入节点与文本内容建模对节点关系建模的指导和影响.

以上过程详细讨论了优化邻节点对网络表示学习过程的影响,又讨论了引入关系模型建模对建模节点与文本之间的关联关系的意义,从而强化网络表示学习过程.这 2 个部分可单独建模和学习,验证其对网络表示学习的影响.为了学习得到更加稳健的网络表示,这 2 个部分可进行联合学习,从而提出了一个统一的联合学习框架.而本文提出的 NRNR 算法正是为了解决以上 3 个问题而提出.本文通过 3 个真实的引文网络数据集验证了其有效性和鲁棒性.

1 相关工作

表示学习又被称为嵌入学习,而 Bengio 等人^[15]于 2003 年首次提出了“Word Embedding”一词.随后, Bengio 等人^[16]在 2013 年提出了一种基于深度神经网络的表示学习算法,但是该算法受限于计算机计算能力的限制并没有得到重视.而表示学习真正地受到科研人员的重视是在 Mikolov 等人^[3-4]与 2013 年提出了 Word2Vec 算法之后. Word2Vec 使用窗口获取当前词语的上下文节点对,然后将节点对输入到一个浅层的 3 层神经网络中,基于节点对的共现不断地调整词语之间的表示向量. Word2Vec 算法被广泛地应用于各类自然语言处理任务^[17-20]. 2014 年 Perozzi 等人^[2]基于 Word2Vec 算法提出了 DeepWalk 算法. DeepWalk 在语言模型中引入了随机游走的采样方法,通过随机游走粒子在网络上的

游走模拟语言模型中的句子,底层同样采用 Word2Vec 所采用的 3 层神经网络模型.DeepWalk 被广泛应用于网络节点分类^[11-12]、链路预测^[21-22]、推荐系统^[23]和可视化^[24]等任务中。

DeepWalk 为大规模网络表示学习任务提供了一种高效的解决方法,但是因其模型简单,在某些任务中效果并没有传统的方法好,因此基于 DeepWalk 提出了各类网络表示学习的改进模型.例如 node2Vec^[5]和 Walklets^[6]改进了网络表示学习中的随机游走过程.LINE^[25]引入 1 阶相似度和高阶相似度对大规模网络进行表示学习.NEU^[26](network embedding update)和 GraRep^[27]提出了高阶逼近的网络表示学习.TADW^[7]仅引入诱导矩阵补全算法^[8]将网络节点的文本特征嵌入到网络表示学习中.DDRW^[28](discriminative deep random walk),MMDW^[29](max-margin deepwalk),TLINE^[30],GENE^[31](group enhanced network embedding)和 SemiNE^[32](semi-supervised network embedding)引入了网络节点的标签类别信息约束网络表示学习模型.TriDNR^[9](tri-party deep network representation),LDE^[33](linked document embedding),DMF^[34](discriminative matrix factorization),Planetoid^[35](predicting labels and neighbors with embeddings transductively or inductively from data),LANE^[36](label informed attributed network embedding)同时引入了网络节点的文本内容和节点的标签类别约束网络表示学习模型.针对动态网络和超网络等特殊网络结构,也提出了相应的网络表示学习算法,例如 DynamicTriad^[37](dynamic triadic),DepthLG^[38],DHNE^[39](deep hyper-network embedding)等.另外,生成对抗网络(generative adversarial network, GAN)^[40]也被引入到网络表示学习中,例如 ANE^[41](adversarial network embedding),GraphGAN^[42],NetGAN^[43]等。

以上内容详细地讨论了网络表示学习的起源,描述和解释了网络表示学习中的经典算法,并介绍了最新的一些网络表示学习算法,例如特殊结构的网络表示学习和生成对抗式的网络表示学习算法等.但是,以上部分算法在基于神经网络训练表示学习模型时,并未考虑不同位置的上下文节点对中心节点的影响,也未考虑使用关系模型构建节点对之间的关系.因此,本文提出的 NRNR 算法正是为了优化以上 2 个目标而提出.因为,在上下文窗口中,不同位置的节点对中心节点的影响力是不相同的。

应该使中心节点的 1 阶邻居节点比 2 阶邻居节点拥有更大的影响力.另外,通过矩阵分解或者异构网络将文本特征嵌入到网络表示具有较大的计算开销,而本文首次采用关系模型机制将及节点之间的多元关系嵌入到网络表示,其计算开销较小且拥有统一的目标函数,且在多元关系向量建模过程中优化具有相同关系节点对之间的表示向量学习过程。

2 算法设计

2.1 定义

本节对本文中常用的一些变量进行定义.本文使用 $G=(V,E)$ 定义一个网络,其中 V 表示网络的节点集, E 表示网络中的边集, $|V|$ 表示网络中节点个数.使用 $\mathbf{R}_v \in \mathbb{R}^k$ 表示 NRNR 训练所得的网络表示向量,该表示向量是一个 k 维的矩阵,矩阵中的每一行表示一个节点 k 维的网络表示向量。

2.2 关系模型

关系模型最经典的算法为 TransE(translating embeddings),该算法于 2013 年被 Bordes 等人^[14]在 NIPS(neural information processing systems)会议上提出.TransE 被提出主要是用来解决大规模知识库中知识的表征学习^[44],即研究如何将知识嵌入到一个低维度的向量空间中,其目的是建模多关系数据.现有的知识库有 OpenCyc(open cycorp),WordNet,Freebase,Dbpedia^[13]等.这些知识库中已有大量的知识,但是这些知识都是基于现有的知识抽取而得,而研究基于知识库的知识推理就需要研究如何对知识进行表征,进而从现有的知识中推导出大量的未知知识。

TransE 是基于翻译机制的知识嵌入模型,其将关系三元组 (h,r,t) 中的关系 r 作为实体 h 到实体 t 的翻译,并通过关系三元组的不断涌现从而持续地调整实体 h 、关系 r 和实体 t 的表示向量,使实体 h 与关系 r 的和向量 $\mathbf{x}_h + \mathbf{x}_r$ 尽可能与实体 t 的向量 \mathbf{x}_t 具有更近的空间距离,即模型训练的目标为 $\mathbf{x}_h + \mathbf{x}_r \approx \mathbf{x}_t$ 基于该目标,TransE 的目标函数为

$$L = \sum_{(h,r,t) \in S} \sum_{(h',r,t') \in S'(h,r,t)} [\lambda + d(\mathbf{x}_h + \mathbf{x}_r, \mathbf{x}_t) - d(\mathbf{x}_{h'} + \mathbf{x}_r, \mathbf{x}_{t'})]_+, \quad (1)$$

其中:

$$S'(h,r,t) = \{(h',r,t) | h' \in E\} \cup \{(h,r,t') | t' \in E\}, \quad (2)$$

$$d(\mathbf{x}_h + \mathbf{x}_r, \mathbf{x}_t) = \|(\mathbf{x}_h + \mathbf{x}_r) - \mathbf{x}_t\|_2^2. \quad (3)$$

S 为三元组 (h, r, t) 的集合, S' 为三元组 (h, r, t) 的负采样集合, 即存在于知识库中的 (h, r, t) 为正样本, 不存在于知识库中的 (h, r, t) 为负样本. $\lambda > 0$ 为间隔距离超参; $x > 0$ 时 $[x]_+ = x$, $x \leq 0$ 时 $[x]_+ = 0$.

本文使用节点关系三元组作为关系模型的输入. 在构建节点关系三元组时, 本文仅考虑了一元关系. 一元关系是 2 个节点的文本中含有一个共同词语时, 则将该词语作为 2 个节点的关系. 例如, 节点 1 的文本标题为“Neural Network for Pattern Recognition”, 节点 2 的文本标题为“Neural Network: A Comprehensive Foundation”. 则可以构建的节点三元组为 $(Node_1, neural, Node_2)$ 和 $(Node_1, network, Node_2)$. 这种类型的节点三元组被称为含一元关系的节点三元组.

2.3 使用负采样的 CBOW 模型

Word2Vec 提供了 2 种模型: CBOW 和 Skip-Gram, 并提供了 2 种优化方法: 负采样和层次化的 Softmax^[3-4]. DeepWalk 是基于 Word2Vec 模型提出的大规模网络表示学习算法^[2], 因此, DeepWalk 完全继承了 Word2Vec 提供的训练模型和优化方法. CBOW 模型训练速度快, 但是精度略劣于 Skip-Gram. 负采样优化方法由于不用构建 huffman 树, 故优化速度也优于层次化的 Softmax. 因此, 本文采用负采样优化的 CBOW 训练 DeepWalk 模型, 并将 1 阶邻节点优化的目标函数和关系模型的目标函数融入到 DeepWalk 的目标函数中.

对于当前中心节点 v , 其上下文定义为 $Context(v)$. 负采样集合定义为 $NEG(v)$, 且 $NEG(v) \neq \emptyset$. 对于 $\forall u \in D$, D 表示节点集合. 定义节点 u 的标签为

$$L^v(u) = \begin{cases} 1, & u = v, \\ 0, & u \neq v, \end{cases} \quad (4)$$

即正样本的标签为 1, 负样本的标签为 0.

在负采样的过程中, 当前中心节点为正样本, 其他所有节点为负样本, 然后将所有概率相乘使其最大化. 对于单个样本 $(v, Context(v))$ 有:

$$p(\xi | Context(v)) = \begin{cases} \sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi), & L^v(\xi) = 1, \\ 1 - \sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi), & L^v(\xi) = 0, \end{cases} \quad (5)$$

则所有样本的概率之和为

$$g(v) = \prod_{\xi \in \{v\} \cup NEG(v)} p(\xi | Context(v)), \quad (6)$$

\mathbf{x}_v 是上下文 $Context(v)$ 中每个节点的向量之和, $\boldsymbol{\theta}^\xi$ 是节点 ξ 的待训练向量, 式(6)可被简化为

$$p(\xi | Context(v)) = [\sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi)] L^v(\xi) \cdot [1 - \sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi)]^{1 - L^v(\xi)}, \quad (7)$$

因此, 函数 $g(v)$ 通过式(5)和式(7)生成为

$$g(v) = \prod_{\xi \in \{v\} \cup NEG(v)} [\sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi)]^{L^v(\xi)} \cdot [1 - \sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi)]^{1 - L^v(\xi)}. \quad (8)$$

综上, 基于语料 C , 负采样优化的 CBOW 的整体优化目标函数可定义为

$$G = \prod_{v \in C} g(v). \quad (9)$$

对式(9)取对数操作, 则整体的目标函数可修改为

$$\begin{aligned} \mathcal{L}(v) &= \lg G = \lg \prod_{v \in C} g(v) = \sum_{v \in C} \lg g(v) = \\ &= \sum_{v \in C} \lg g(v) - \prod_{\xi \in \{v\} \cup NEG(v)} [\sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi)]^{L^v(\xi)} \cdot \\ &= [1 - \sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi)]^{1 - L^v(\xi)} = \\ &= \sum_{v \in C} \sum_{\xi \in \{v\} \cup NEG(v)} \{L^v(\xi) \cdot \lg[\sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi)] + \\ &= (1 - L^v(\xi)) \lg[1 - \sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi)]\}. \end{aligned} \quad (10)$$

在神经网络模型中, 目标函数一般取对数似然函数, 且对数的底通常省略, 因为在最终的参数更新公式中, 对数形式会被转化为非对数形式. 从式(10)抽取基于负采样优化的 CBOW 模型的目标函数最常用的形式为

$$\mathcal{L}(v) = \sum_{v \in C} \sum_{\xi \in \{v\} \cup NEG(v)} \lg p(\xi | Context(v)) = \sum_{v \in C} \lg g_{C_v}(v), \quad (11)$$

其中:

$$g_{C_v}(v) = \prod_{\xi \in \{v\} \cup NEG(v)} p(\xi | Context(v)). \quad (12)$$

式(11)中, C 为节点序列语料, 使用随机梯度上升法获得式(10)中变量 $\boldsymbol{\theta}^\xi$ 的更新公式为

$$\begin{aligned} \boldsymbol{\theta}^\xi &:= \boldsymbol{\theta}^\xi + \mu \cdot [L^v(\xi) - \sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi)] \cdot \mathbf{x}_v, \\ c(v) &:= c(v) + \end{aligned} \quad (13)$$

$$\mu \cdot \sum_{\xi \in \{v\} \cup NEG(v)} [L^v(\xi) - \sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi)] \cdot \boldsymbol{\theta}^\xi. \quad (14)$$

式(13)中 μ 表示学习率, \mathbf{x}_v 为 $Context(v)$ 中每个节点的表示向量之和. 通过 \mathbf{x}_v 可以获得 $Context(v)$ 中每个节点的表示向量更新方式, 如式(14)所示.

2.4 NRNR 建模

本文提出的 NRNR 算法旨在解决如何使用 1 阶邻节点优化网络表示学习过程, 同时解决如何采用网络节点的文本内容优化网络表示学习过程. 为了适应大规模网络表示学习任务要求, 需要提出一种简单高效的联合学习模型. 为了实现该方案,

本文提出了一种基于神经网络的网络表示联合学习框架,该框架由 3 个部分构成,分别为 1 阶邻节点优

化部分、网络节点关系建模部分和节点与文本关系模型构建部分.具体如图 1 所示:

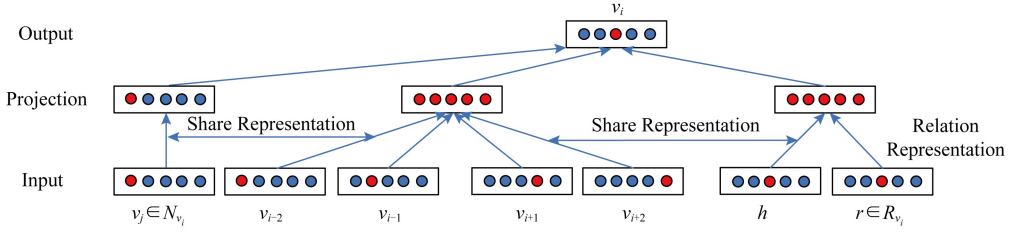


Fig. 1 NRNR algorithm framework

图 1 NRNR 算法框架

如图 1 所示, N_{v_i} 表示当前中心节点 v_i 的 1 阶邻节点集合, R_{v_i} 表示当前中心节点 v_i 的节点三元组集合.在 Word2Vec 中,CBOW 模型是通过上下文词语出现的概率去预测当前中心词出现的概率.而 DeepWalk 算法是基于 Word2Vec 算法的改进,因此,底层同样可以采用 CBOW 模型学习节点与节点之间的关系.例如对于当前中心节点 v_i ,CBOW 模型通过其在随机游走序列中的前面 2 个节点 v_{i-2} 和 v_{i-1} 以及其后面 2 个节点 v_{i+1} 和 v_{i+2} 来预测当前中心节点 v_i 出现的概率.CBOW 模型通过不断地调整网络表示向量中的值,使得具有连边的节点对之间具有更近的代表向量空间距离,使得具有多跳边或者没有边的节点对之间具有较远的代表向量空间距离.本文提出的 NRNR 算法在该调节的过程中添加了一些约束,即使得当前中心节点 v_i 的 1 阶邻居节点 v_{i-1} 和 v_{i+1} 比 2 阶邻居节点 v_{i-2} 和 v_{i+2} 在向量表示空间中与 v_i 具有较近的空间距离.在图 1 中, $v_j \in N_{v_i} = (v_{i-1}, v_{i+1})$.另外,NRNR 算法添加了文本特征,该文本特征是将具有共同出现词语的节点对之间转化为节点三元组关系.NRNR 算法采用多源关系建模思想优化了网络表示向量学习过程,即使得具有三元组关系约束的节点对比没有三元组关系约束的节点对在网络表示向量空间中具有更近的空间距离.

在 NRNR 算法中,以上过程可被认为是 3 个 CBOW 模型的叠加.另外,CBOW 模型与 1 阶邻节点优化过程共享一份节点向量,同时,与关系模型优化过程也共享 1 份节点向量.因此,本文提出的 NRNR 通过共享向量来获得和交换彼此的特征信息,使得 NRNR 算法在建模学习过程中能够从 1 阶邻居节点和节点三元组中获取有价值的特征信息,从而使获得的网络表示能够在各类任务中具有更强的泛化能力.另外,共享节点向量为联合学习模型提供了解决思路.

为了将 1 阶邻节点信息融入到网络表示学习模型中,本文提出的目标函数为

$$\mathcal{L}_{\text{NRNR_N}} = \sum_{v \in C} (\text{lb } g_{C_v}(v) + \alpha \cdot \sum_{u \in N_v} \text{lb } g_{N_v}(u)). \quad (15)$$

本文中,NRNR_N 算法的学习目标是最大化式 (15),因此,本文采用了类似于 Word2Vec^[3-4] 所采用的随机梯度上升法获得每个参数的更新公式.其中, $g_{C_v}(v)$ 与 $g_{N_v}(v)$ 的具有相同的形式.式 (15) 的 v 即为图 1 中的 v_i ,因为我们在式 (15) 中最外层的求和符号下标设定为 $v \in C$,如果将此下标设置为 $1 \leq i \leq |C|$,则式 (15) 中的 v 全部应替换为 v_i .式 (15) 的左边项为 CBOW 的目标函数,其参数更新见式 (13)(14).式 (15) 右边项为本文添加的约束项,其本质为一个 CBOW 模型,即再一次使用 1 阶邻居节点优化网络表示学习模型,故优化求解方法与 CBOW 完全相同.首先我们设求偏导的表达式为

$$F_1 = \alpha \cdot \sum_{v \in C} \sum_{u \in N_v} \text{lb } g_{N_v}(u) = \alpha \cdot \sum_{v \in C} \sum_{u \in N_v} \sum_{\xi \in \{v\} \cup \text{NEG}(u)} \{L^\xi(u) \cdot \text{lb}[\sigma(\mathbf{x}_u^T \boldsymbol{\theta}^\xi)] + (1 - L^\xi(u)) \text{lb}[1 - \sigma(\mathbf{x}_u^T \boldsymbol{\theta}^\xi)]\}. \quad (16)$$

令:

$$f_1 = L^\xi(u) \cdot \text{lb}[\sigma(\mathbf{x}_u^T \boldsymbol{\theta}^\xi)] + (1 - L^\xi(u)) \text{lb}[1 - \sigma(\mathbf{x}_u^T \boldsymbol{\theta}^\xi)]. \quad (17)$$

式 (16) 中,对于当前节点的 1 阶邻居节点集合而言,目标节点是正样本,而该集合之外的所有节点均为负样本.然后采用 f_1 分别对 $\boldsymbol{\theta}^\xi$ 和 \mathbf{x}_u 求偏导.最终求得式 (17) 中 $\boldsymbol{\theta}^\xi$ 和 1 阶邻居节点 $\mathbf{n}(u)$ 的更新公式为

$$\boldsymbol{\theta}^\xi := \boldsymbol{\theta}^\xi + \mu \cdot [L^\xi(u) - \sigma(\mathbf{x}_u^T \boldsymbol{\theta}^\xi)] \cdot \mathbf{x}_u, \quad (18)$$

$$\mathbf{n}(u) := \mathbf{n}(u) + \mu \cdot$$

$$\sum_{\xi \in \{v\} \cup \text{NEG}(v)} [L^\xi(u) - \sigma(\mathbf{x}_u^T \boldsymbol{\theta}^\xi)] \cdot \boldsymbol{\theta}^\xi. \quad (19)$$

本文中通过权重 α 控制 1 阶邻居的权重,因此权重 α 需要乘到式 (18)(19) 中的 μ 之前.另外, \mathbf{x}_u

为当前节点的 1 阶邻居节点的表示向量之和, $\mathbf{n}(u)$ 为其中每个 1 阶邻居节点的表示向量。

NRNR 然引入了关系模型, 但是并未使用 TransE^[14] 算法建模节点三元组, 如果使用 TransE 模型, 需要将 TransE 的目标函数融入到 DeepWalk 算法的目标函数中. 在 TransE 算法中, 三元组 (h, r, v) 满足 $\mathbf{x}_h + \mathbf{x}_r \approx \mathbf{x}_v$. 该式子是 TransE 算法的训练目标, 即如果节点 v 和节点 h 之间存在关系 r , 则让节点 h 和节点 r 的向量之和尽量靠近节点 v 的表示向量, 否则应该远离 v 的表示向量. 本文将关系模型的思想引入到 NRNR 算法中, 实现了在建模网络节点对之间关系的同时建模了节点三元组之间的关系. 基于此, 本文提出了一种基于 CBOW 的关系模型构建方法 NRNR_R. 其目标函数为

$$\mathcal{L}_{\text{NRNR_R}} = \sum_{v \in C} (\text{lb } g_{C_v}(v) + \beta \cdot \sum_{r \in R_v} \text{lb } g_{h+r}(v)). \quad (20)$$

在式(20)中, R_v 表示构成 v 的关系三元组 (h, r, v) 的集合. 式(20)中 $g_{h+r}(v)$ 可被认为是通过节点 h 之间的关系 r 来预测节点 v 出现的概率. 式(20)类似于式(15), 均可被认为是 2 个 CBOW 模型的叠加, 左边的第 1 个 CBOW 模型的更新公式可见式(13)(14). 右边第 2 项我们设:

$$F_2 = \beta \cdot \sum_{v \in C} \sum_{r \in R_v} \text{lb } g_{h+r}(v) = \beta \cdot \sum_{v \in C} \sum_{r \in R_v} \sum_{\xi \in \{v\} \cup \text{NEG}(v)} \{L^v(\xi) \cdot \text{lb}[\sigma(\mathbf{x}_{h+r}^T \boldsymbol{\theta}^\xi)] + (1 - L^v(\xi)) \text{lb}[1 - \sigma(\mathbf{x}_{h+r}^T \boldsymbol{\theta}^\xi)]\}. \quad (21)$$

令:

$$g_{h+r}(v) = \prod_{\xi \in \{v\} \cup \text{NEG}(v)} [\sigma(\mathbf{x}_{h+r}^T \boldsymbol{\theta}^\xi)]^{L^v(\xi)} \cdot [1 - \sigma(\mathbf{x}_{h+r}^T \boldsymbol{\theta}^\xi)]^{1-L^v(\xi)}. \quad (22)$$

又令:

$$f_2 = L^v(\xi) \cdot \text{lb}[\sigma(\mathbf{x}_{h+r}^T \boldsymbol{\theta}^\xi)] + (1 - L^v(\xi)) \text{lb}[1 - \sigma(\mathbf{x}_{h+r}^T \boldsymbol{\theta}^\xi)]. \quad (23)$$

NEG(v) 进行负采样时, 客观存在的三元组设置为正样本, 而非客观存在的三元组设置为负样本. 然后用函数 f_2 对 $\boldsymbol{\theta}^\xi$ 求偏导数, 最终得到 $\boldsymbol{\theta}^\xi$ 的更新公式为

$$\boldsymbol{\theta}^\xi := \boldsymbol{\theta}^\xi + \mu \cdot [L^v(\xi) - \sigma(\mathbf{x}_{h+r}^T \boldsymbol{\theta}^\xi)] \cdot \mathbf{x}_{h+r}. \quad (24)$$

由于 $\mathbf{x}_{h+r} = \mathbf{x}_h + \mathbf{x}_r$, 因此, 需要单独对 \mathbf{x}_h 和 \mathbf{x}_r 求偏导, 得到最终的更新公式为

$$\mathbf{x}_h := \mathbf{x}_h + \mu \cdot \sum_{\xi \in \{v\} \cup \text{NEG}(v)} [L^v(\xi) - \sigma(\mathbf{x}_{h+r}^T \boldsymbol{\theta}^\xi)] \cdot \boldsymbol{\theta}^\xi, \quad (25)$$

$$\mathbf{x}_r := \mathbf{x}_r + \mu \cdot \sum_{\xi \in \{v\} \cup \text{NEG}(v)} [L^v(\xi) - \sigma(\mathbf{x}_{h+r}^T \boldsymbol{\theta}^\xi)] \cdot \boldsymbol{\theta}^\xi. \quad (26)$$

本文中通过权重 β 控制三元组的权重, 因此, 权重 β 需要乘到式(24)~(26)中的 μ 之前。

$\mathcal{L}_{\text{NRNR_N}}$ 和 $\mathcal{L}_{\text{NRNR_R}}$ 分别实现了 1 阶邻节点的学习过程和节点三元组被转化为网络表示的过程. 为了将这 2 个优化方法同时融入到网络表示学习过程中, 本文提出了带 1 阶邻节点优化和节点三元组优化的网络表示学习算法 NRNR_NR, 其目标函数为

$$\mathcal{L}_{\text{NRNR_N}} = \sum_{v \in C} (\text{lb } g_{C_v}(v) + \alpha \times \sum_{u \in N_v} \text{lb } g_{N_v}(u) + \beta \times \sum_{r \in R_v} \text{lb } g_{h+r}(v)). \quad (27)$$

式(27)可被认为是 3 个 CBOW 模型的叠加, 因此, 每个部分的参数更新即为整个表达式的参数更新公式. 故式(27)的参数更新公式由式(13)(14)(18)(19)(24)~(26)组成. 另外, 也需要在参数更新公式中添加权重 α 和 β . 为了更加详尽地介绍本文提出的 NRNR 算法的具体流程, 我们给出了算法伪代码:

算法 1. NRNR.

输入: 图 $G(V, E)$ 、节点向量大小 d 、每个节点的随机游走数量 wp 、随机游走长度 wl 、节点三元组 R 、节点三元组权重 β 、1 阶邻居节点权重 α ;

输出: 节点向量 \mathbf{V} .

/* 获取所有节点的随机游走序列 */

① for $i=0$ to wp do

② $O \leftarrow \text{Shuffle}(V)$;

③ for $v \in O$ do

④ $C \leftarrow \text{WalkSeqAppend}(G, v, wl)$;

⑤ end for

⑥ end for

/* 初始化所有参数 */

⑦ $vertex_size \leftarrow \text{GetVocabSize}(C)$;

⑧ $relation_num \leftarrow \text{GetRelationNum}(R)$;

⑨ $\mathbf{V} \leftarrow \text{InitVector}(vertex_size, d)$;

⑩ $\boldsymbol{\theta} \leftarrow \text{InitVector}(vertex_size, d)$;

⑪ $\mathbf{P} \leftarrow \text{InitVector}(relation_num, d)$;

/* 对每一个节点作如下训练和优化 */

⑫ for $v \in C$ do

/* 训练原始的负采样优化的 CBOW 模型 */

⑬ $e \leftarrow 0$;

⑭ $\mathbf{x}_v \leftarrow \sum_{\xi \in \text{Context}(v)} \mathbf{V}(\xi)$;

/* 根据式(13)(14)更新 CBOW 模型的参数 */

⑮ for $\xi \in \{v\} \cup \text{NEG}(v)$ do

⑯ $q \leftarrow \sigma(\mathbf{x}_v^T \boldsymbol{\theta}^\xi)$;

17 $g \leftarrow \mu \cdot [L^v(\xi) - q];$

18 $e \leftarrow e + g\theta^\xi;$

19 $\theta^\xi \leftarrow \theta^\xi + gx_v;$

20 end for

/* 更新上下文节点的表示向量 */

21 for $\xi \in Context(v)$ do

22 $V(\xi) \leftarrow V(\xi) + e;$

23 end for

/* 采用 1 阶邻居节点优化网络表示学习模型 */

24 for $u \in N_v$ do

25 $e \leftarrow 0;$

/* 根据式(18)(19)更新模型参数 */

26 for $\xi \in \{v\} \cup NEG(u)$ do

27 $q \leftarrow \sigma(x_u^T \theta^\xi);$

28 $g \leftarrow \mu \times [L^\xi(u) - q];$

29 $e \leftarrow e + g\theta^\xi;$

30 $\theta^\xi \leftarrow \theta^\xi + agx_u;$

31 end for

32 $V(\xi) \leftarrow V(\xi) + ae;$

33 end for

/* 采用节点三元组优化网络表示学习模型 */

34 for $r \in R_v$ do

35 $e \leftarrow 0;$

/* 获取所有的节点三元组 */

36 $(h, r, v) \leftarrow GetTriplet(R, r, v);$

/* 根据式(24)~(16)更新模型参数 */

37 for $\xi \in \{v\} \cup NEG(v)$ do

38 $x_{h+r} \leftarrow V(v) + P(v);$

39 $q \leftarrow \sigma(x_{h+r}^T \theta^\xi);$

40 $g \leftarrow \mu \cdot [L^v(\xi) - q];$

41 $e \leftarrow e + g\theta^\xi;$

42 $\theta^\xi \leftarrow \theta^\xi + \beta gx_{h+r};$

43 end for

44 $V(v) \leftarrow V(v) + \beta e;$

45 $P(v) \leftarrow P(v) + \beta e;$

46 end for

47 end for

3 实验与分析

3.1 数据集描述

本文使用 Citeseer(M10),DBLP(V4)数据集验证本文提出算法的可行性.为了验证在稠密网络中的可行性,本文基于 DBLP(V4)构造了一个高平均度的网络 Shifted DBLP(SDBLP).在 SDBLP 中删除了所有孤立节点,并且删除了节点连边数少于或等于 2 的所有节点及其连边.3 个数据集的具体属性描述如表 1 所示:

Table 1 Property Description on Citeseer, DBLP and SDBLP Datasets
表 1 Citeseer,DBLP 和 SDBLP 数据集属性描述

Dataset	Original Vertex Number	Original Edge Number	Separate Vertex Number	Remaining Vertex Number	Remaining Edge Number	Average Clustering Coefficient	Average Degree	Number of Triplets	Number of Relations
Citeseer	10 310	5 923	5 700	4 610	5 923	0.264	2.57	1 263 566	6 524
DBLP	60 744	105 781	43 019	17 725	105 781	0.187	11.936	15 848 328	11 858
SDBLP	60 744	105 781	0	3 119	39 516	0.259	25.339	510 793	4 451

如表 1 所示,列 2 和列 3 分别为原始数据集中的节点数量和连边数量.列 4 为网络中孤立节点的数量.经过删除孤立节点等操作,剩余的节点数量和连边数量如列 5 和列 6 所示.最终删除后的网络中,Citeseer,DBLP,SDBLP 数据集中节点个数分别为 4610,17725,3119,网络平均度分别为 2.57,11.936,25.339.可以发现,网络的密度越来越高.本文使用 Citeseer,DBLP,SDBLP 模拟 3 类属性不同的网络数据集,从而验证本文提出算法的泛化能力.

3.2 对比算法

1) DeepWalk.DeepWalk 算法是基于神经网络

的最经典的网络表示学习算法,后续的诸多网络表示学习算法均是基于 DeepWalk 算法而提出.DeepWalk 起源于 Word2Vec 算法.DeepWalk 可使用 CBOW 和 Skip-Gram 两种模型训练基于神经网络的表示学习模型,另可采用负采样和层次化的 Softmax 加速网络训练过程,CBOW 具有训练速度快的特点,Skip-Gram 具有精度高的特点.在本文中,使用 CBOW 和负采样训练 DeepWalk.

2) LINE.Tang 等人^[25]提出了一种大规模网络表示学习算法,该算法通过部分舍弃精度的方式追求在超大规模网络中编码网络结构成为低维度的网

络表示.因此,LINE 的训练速度非常快,但是精度不高,尤其在稀疏网络中网络节点分类性能较差.这种速度的提升主要来自于 LINE 仅考虑网络的 1 阶相似度 1st LINE 或者 2 阶相似度 2nd LINE.本文中 使用 2nd LINE 训练网络表示.

3) node2Vec.node2Vec 是基于 DeepWalk 算法而提出的一种网络表示学习算法.DeepWalk 的随机游走策略是完全的随机,而 node2Vec 改进了 DeepWalk 的随机游走策略,即采用了图论中的广度优先搜索策略和深度优先搜索策略来采样节点,广度优先搜索控制网络的全局宏观视图特征,深度优先搜索控制网络的局部微观视图特征.

4) GraRep.GraRep 是一种基于矩阵分解的高阶网络表示学习算法,在该算法中,定义 1 阶的概率转移矩阵为 $A=D^{-1}S$,其中 S 为邻接矩阵, D 为对角矩阵,则第 k 阶的网络结构特征定义为 A^k .然后使用 SVD 去分解每一阶的网络结构特征,最后拼接所有阶的表示向量.在本文中,设置 $k=3$.

5) Text Feature(TF).文本特征是将网络节点的文本内容转化为共现矩阵,然后使用 SVD 去分解该共现矩阵,从而得到一个列维度为 100 的文本特征向量.本文使用该文本特征 $R_v \in \mathbb{R}^{|V| \times 100}$ 作为节点的表示向量.

6) DeepWalk+TF.将 DeepWalk 和 TF 生成的网络表示向量通过列向量扩充的形式拼接在一起.

7) MFDW.网络表示学习算法 DeepWalk 被证明是分解矩阵 $M=(A+A^2)/2$,本文使用 SVD 算法去分解矩阵 M ,并使用 $W=U \cdot S^{0.5}$ 作为网络的表示向量.

8) TADW.网络表示学习算法 DeepWalk 被证明是分解矩阵 $M=(A+A^2)/2$,其中 A 为邻接矩阵. TADW 并非使用 SVD 去分解矩阵 M ,而是引入了诱导矩阵补全算法去分解矩阵 M .因为诱导矩阵补全算法可提供一个外部矩阵辅助分解矩阵 M 的功能.

9) STADW.该算法是 TADW 算法的简化形式,在 TADW 算法中删除了对文本特征的优化处理.

3.3 实验设置

本文使用网络节点分类任务评估本文提出的算法与本文引入的对比算法.本文使用 Liblinear^[45] 作为基线分类器.为了验证算法的泛化能力,本文将训练集设置为 0.1~0.9,共 9 个比例的训练集,将剩余的网络节点作为测试集.网络表示学习算法得到的网络表示向量统一设置为 100 维.另外,设置随机游走长度为 40,随机游走个数为 10 条,窗口大小为 5,负采样为 5,最小节点频度为 5,神经网络的学习率为 0.05,1 阶邻居节点的权重为 0.7,节点三元组的权重为 0.3.本文中所有实验均重复 10 次取平均值作为最终的结果.

3.4 实验结果分析

本文使用 Citeseer,DBLP,SDBLP 等真实的网络数据集作为评估数据集.并从数据集中抽取 10%~90%的数据作为训练集,剩余的数据作为测试集.表 2~4 列出了本文引入的对比算法和本文提出的算法在 3 个数据集和 9 种训练集比例下的网络节点分类准确率.

Table 2 Accuracy of Vertex Classification on Citeseer
表 2 Citeseer 数据集上的节点分类准确率

Algorithm	The Proportions of Dataset								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
DeepWalk	0.558 9	0.593 0	0.608 9	0.614 8	0.621 9	0.623 0	0.626 2	0.623 3	0.639 5
LINE	0.426 4	0.470 6	0.480 4	0.495 7	0.504 3	0.510 2	0.511 8	0.530 7	0.536 3
node2Vec	0.624 7	0.656 1	0.660 0	0.670 7	0.674 0	0.671 5	0.674 6	0.680 7	0.685 6
GraRep	0.393 8	0.530 9	0.578 5	0.597 5	0.599 7	0.610 5	0.615 7	0.620 9	0.608 9
Text Feature(TF)	0.576 9	0.613 0	0.627 6	0.630 5	0.634 8	0.633 0	0.628 7	0.621 9	0.639 5
DeepWalk+TF	0.583 1	0.611 5	0.627 3	0.633 7	0.641 8	0.639 6	0.655 0	0.654 9	0.653 0
MFDW	0.576 2	0.607 9	0.623 3	0.630 5	0.629 6	0.630 0	0.630 0	0.634 8	0.643 0
TADW	0.684 7	0.722 9	0.735 0	0.742 5	0.742 4	0.747 3	0.753 6	0.749 5	0.754 7
STADW	0.467 1	0.619 9	0.659 9	0.716 0	0.731 1	0.738 2	0.733 3	0.742 7	0.743 6
NRNR_N	0.568 9	0.606 1	0.617 9	0.624 7	0.630 4	0.642 4	0.644 9	0.647 9	0.653 5
NRNR_R	0.746 9	0.767 8	0.774 9	0.782 6	0.781 8	0.788 1	0.788 8	0.794 3	0.795 2
NRNR_NR	0.764 7	0.783 3	0.792 9	0.795 1	0.799 7	0.801 2	0.801 3	0.809 8	0.810 3

Table 3 Accuracy of Vertex Classification on DBLP

表 3 DBLP 数据集上的节点分类准确率

Algorithm	The Proportions of Dataset								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
DeepWalk	0.622 6	0.643 4	0.654 2	0.659 8	0.662 4	0.661 8	0.666	0.670 3	0.667 7
LINE	0.644 9	0.665 3	0.674 9	0.678 7	0.679 8	0.683 0	0.690 3	0.688 9	0.688 6
node2Vec	0.733 9	0.739 8	0.752 5	0.756 1	0.757 0	0.758 5	0.757 9	0.757 3	0.763 6
GraRep	0.589 0	0.659 0	0.672 6	0.679 2	0.687 7	0.688 8	0.692 6	0.695 6	0.697 9
Text Feature (TF)	0.661	0.694 6	0.704 9	0.711 5	0.712 9	0.714 4	0.715 4	0.715 7	0.718 3
DeepWalk+TF	0.626 1	0.651 5	0.659 9	0.662 2	0.663 7	0.666 0	0.670 3	0.669 1	0.676 1
MFDW	0.650 2	0.746 8	0.748 8	0.750 2	0.750 5	0.751 3	0.752 2	0.745 7	0.755 1
TADW	0.802 2	0.814 1	0.820 5	0.823 6	0.827 7	0.828 8	0.827 1	0.831 0	0.829 7
STADW	0.753 9	0.805 1	0.810 9	0.810 0	0.816 4	0.817 0	0.817 1	0.816 9	0.813 4
NRNR_N	0.640 6	0.640 4	0.657 9	0.658 4	0.673 2	0.681 2	0.687 8	0.696 2	0.692 1
NRNR_R	0.812 7	0.814 8	0.823 8	0.834 9	0.839 8	0.838 6	0.844 9	0.849 2	0.841 8
NRNR_NR	0.828 7	0.823 9	0.840 6	0.847 4	0.850 2	0.849 6	0.857 6	0.855 6	0.850 7

Table 4 Accuracy of Vertex Classification on SDBLP

表 4 SDBLP 数据集上的节点分类准确率

Algorithm	The Proportions of Dataset								
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
DeepWalk	0.797 6	0.806 5	0.818 8	0.814 9	0.825 6	0.823 5	0.827 3	0.827 1	0.833 7
LINE	0.737 9	0.770 1	0.781 1	0.782 8	0.793 1	0.789 7	0.796 3	0.788 2	0.787 7
node2Vec	0.821 7	0.828 7	0.830 9	0.845 1	0.844 5	0.840 1	0.848 4	0.847 3	0.849 8
GraRep	0.809 9	0.825 2	0.841 4	0.847 8	0.849 7	0.841 7	0.853 6	0.852 7	0.849 5
Text Feature (TF)	0.650 3	0.712 3	0.726 4	0.738 6	0.745 4	0.750 7	0.751 4	0.760 0	0.753 3
DeepWalk+TF	0.793 9	0.809 5	0.816 0	0.814 4	0.825 5	0.822 2	0.821 0	0.825 8	0.831 5
MFDW	0.797 9	0.830 8	0.843 8	0.841 2	0.845 3	0.842 9	0.847 0	0.845 5	0.845 3
TADW	0.813 8	0.831 3	0.835 4	0.848 4	0.840 9	0.843 4	0.850 0	0.856 9	0.857 8
STADW	0.806 3	0.819 2	0.825 4	0.836 0	0.840 2	0.849 2	0.850 3	0.846 5	0.852 7
NRNR_N	0.804 6	0.801 7	0.811 4	0.818 6	0.827 1	0.833 7	0.836 2	0.839 6	0.834 6
NRNR_R	0.814 1	0.828 1	0.831 8	0.841 3	0.840 5	0.844 1	0.850 9	0.853 3	0.850 5
NRNR_NR	0.822 5	0.829 4	0.839 7	0.844 9	0.845 6	0.846 7	0.856 4	0.857 9	0.852 4

如表 2 所示,LINE 算法的网络节点分类性能最差.MFDW 为 DeepWalk 的矩阵分解形式,其在各比例的训练集条件下获得的网络节点分类性能均优于 DeepWalk 算法.DeepWalk 和 node2Vec 算法的底层均为一个 3 层的浅层神经网络算法,不同的是上层的随机游走序列获取方式不同.从实验结果可知,node2Vec 算法的网络节点分类性能优于 DeepWalk 算法.在 Citeseer 数据集上,网络节点的文本特征在节点分类任务中其性能优于 DeepWalk 算法.因此通过简单地拼接策略和类似于 TADW 算法融入文本特征进入网络表示,其获得的网络节点

分类性能均得到了极大地改善.STADW 中删除了对文本特征的优化,因此其性能劣于 TADW.本文提出的 NRNR_N 优化了随机游走过程中的 1 阶邻居节点,其性能优于 DeepWalk 算法.本文提出的 NRNR_R 通过节点三元组的形式将文本特征融入到网络表示中,其性能优于 TADW 算法.本文提出的 NRNR_NR 将 NRNR_N 和 NRNR_R 的思路融合在一起,即同时将 1 阶邻居节点的优化和文本特征融入到网络表示中,其获得的性能优于 NRNR_N 和 NRNR_R,也均优于本文中提出的其他对比算法.

如表 3 所示,在 DBLP 稠密网络上,DeepWalk

的网络节点分类性能略劣于 LINE 和 GraRep. 由于稠密网络中存在大量的连边, 因此, 网络节点之间的结构关系可得到充分地挖掘, node2Vec, DeepWalk 等基于神经网络的网络表示学习算法能够得到充分地训练. 在该数据集中, 同样存在基于 DeepWalk 的矩阵分解 MFDW 算法其分类性能优于 DeepWalk 的现象. 随着训练集比例的提高, 文本特征 TF 的网络节点分类性能越来越优于 DeepWalk. 但是通将 TF 的特征向量与 DeepWalk 的表示向量进行简单拼接, 其获得的性能却劣于文本特征的性能. 通过矩阵分解思想融入文本特征 (TADW), 其网络表示分类性能远优于 DeepWalk, node2Vec, GraRep 等网络表示学习算法. 由于是稠密网络, 因此优化 1 阶邻居节点得到的 NRNR_N 算法其分类性能略优于 DeepWalk. NRNR_R 引入知识表示的思想进入网络表示, 从而将文本特征以三元组约束的形式融入到网络表示中, 其分类性能优于 TADW. NRNR_NR 融合了 NRNR_N 与 NRNR_R 的学习过程, 因此与本文引入的对比算法相比, NRNR_NR 获得了最好的性能.

如表 4 所示, 在 SDBLP 数据集上, 基于矩阵分解的网络表示取得了很好的节点分类性能, 例如, DeepWalk 的矩阵分解算法 MFDW, 在各训练集比例下, 其性能均优于基于神经网络的 DeepWalk 算法. 基于矩阵分解的高阶表示学习 GraRep 算法同样也获得了优于 DeepWalk 的性能. 基于本文提出的 NRNR_N 算法优化了 DeepWalk 算法在随机游走过程中的 1 阶邻居节点, 因此, 其节点分类性能优于 DeepWalk. SDBLP 数据集上, 文本特征的节点分类性能是最差的, 因此, 将 DeepWalk 和文本特征的向量简单拼接后, 其性能的提升是非常有限的. TADW 通过诱导矩阵补全融入文本特征到网络表示, NRNR_R 通过神经网络融入文本特征到网络表示, STADW 删除了 TADW 中对文本特征的优化. 通过实验发现, 在 SDBLP 数据集上, TADW, STADW, NRNR_N 在节点分类任务上的性能几乎相同. 相较于 NRNR_N 和 NRNR_R, NRNR_NR 的性能得到了显著的提升.

在 Citeseer 和 DBLP 数据集上, NRNR_N 算法的网络节点分类性能不如 NRNR_R 和 NRNR_NR 算法. 在 Citeseer 数据集上, NRNR_N 算法的平均分类准确率为 0.626 3, DeepWalk 算法的平均分类准确率为 0.612 2. 在 DBLP 数据集上, NRNR_N 算法的平均分类准确率为 0.669 7, DeepWalk 算法的

平均分类准确率为 0.656 5. 因此, 在 Citeseer 和 DBLP 数据集上, DeepWalk 和 NRNR_N 算法之间的平均分类准确率提升为 2.32% 和 2.02%. 而 NRNR_R 和 NRNR_NR 的提升率远大于该值. 主要原因是虽然 NRNR_N 在网络表示学习过程中优化了 1 阶邻居节点, 使得 1 阶邻居节点与中心节点之间的关联度大于 2 阶邻居节点与中心节点之间的关联度. 但是使用 Citeseer, DBLP, SDBLP 数据集和网络节点分类准确率衡量算法性能时, NRNR_N 和 DeepWalk 算法均能够较好地识别不同类别节点内部节点的标签, 而 NRNR_N 算法能够优化不同类别之间的边界节点被错误分类的过程. 例如在边界节点中 2 阶邻居节点是另外一个类别的节点时, DeepWalk 算法会使得该不同类别的节点与中心节点之间具有较近的表示向量空间距离, 而 NRNR_N 算法仅优化 1 阶邻居节点, 因此能够增进 1 阶邻居节点与中心节点之间的表示向量空间距离, 从而疏远了中心节点与 2 阶邻居节点之间的表示向量空间距离, 进而避免将不同类别的节点分类到同一个类别中. 在 Citeseer 和 DBLP 数据集上, 仅使用文本特征进行网络节点分类, 则文本特征的网络节点分类性能优于 DeepWalk 算法, 因此, NRNR_R 引入了外部文本特征后能够优化网络节点分类性能, 使得 NRNR_R 的网络节点分类性能优于 NRNR_N 和 DeepWalk 等算法.

本文主要提出了 3 种网络表示学习框架: NRNR_N, NRNR_R, NRNR_NR. NRNR_N 是对 DeepWalk 中随机游走序列中的 1 阶邻居节点进行优化, NRNR_R 基于 DeepWalk 模型引入文本特征, NRNR_NR 是 NRNR_N 与 NRNR_R 的融合, TADW 通过矩阵分解思想引入网络节点的文本特征. 因此, 为了直观地展示本文引入的 3 种模型是否有效, 本文图示化了 NRNR_N, NRNR_R, NRNR_NR, DeepWalk, TADW 在 Citeseer, DBLP, SDBLP 上的网络节点分类性能, 具体结果如图 2 所示.

如图 2 所示, 5 种算法在 Citeseer 和 DBLP 数据集上的网络节点分类准确率值跨度大于其在 SDBLP 数据集上的值. 在 SDBLP 数据集上, 5 类对比算法的网络节点分类准确率呈现出了显著的上升趋势. 但在 Citeseer 和 DBLP 数据集上, 准确率曲线表现出了比较缓慢的上升趋势. 另外, 在这 2 类数据集上, DeepWalk 和 NRNR_N 的准确率曲线与 NRNR_R, NRNR_NR, TADW 这 3 条准确率权限

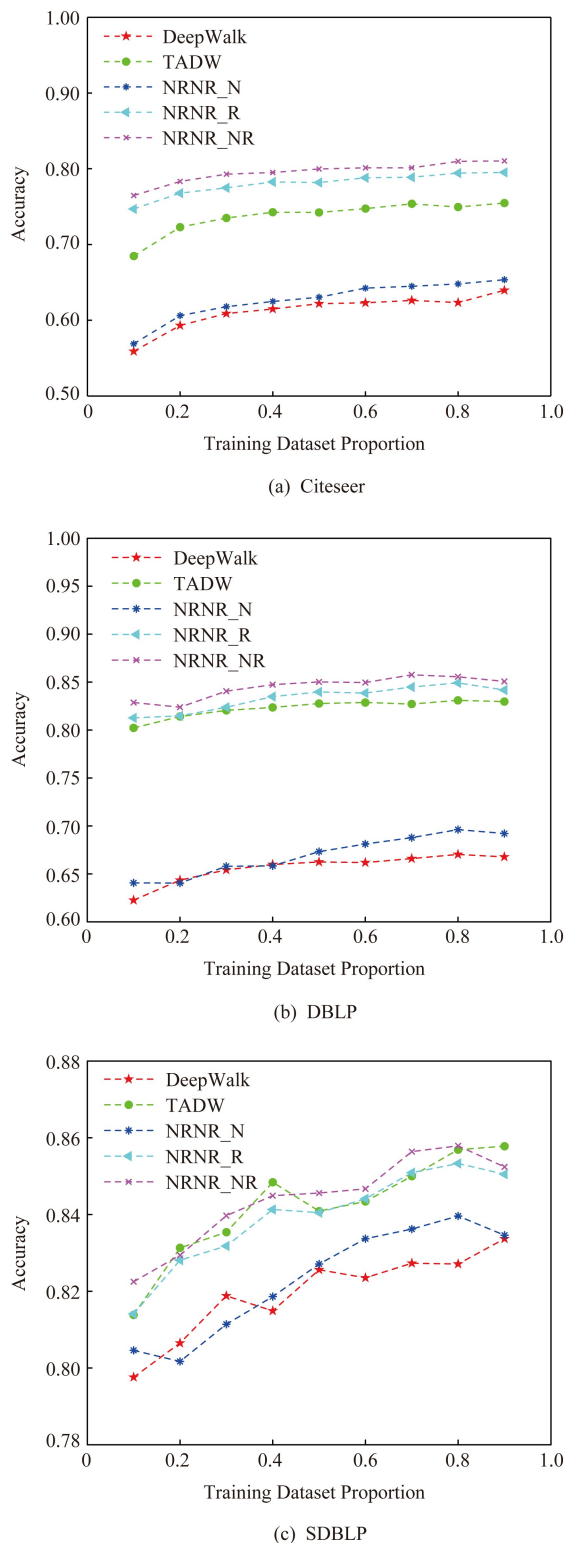


Fig. 2 Performance comparisons of five algorithms on three datasets

图2 3种数据集上5种算法性能对比图

之间有着明显的大跨度.出现这种现象的主要原因是 在稀疏网络上,不同算法之间的获取的特征差异性较大,如果某种算法能够获取较为充分地反映网

络结构的特征,则其网络分类性能变得越好.且在这类稀疏网络上,基于联合学习的网络表示学习算法能够弥补因为边稀疏而带来的训练不充分问题.而在稠密网络上,不同算法均能从充分地边连接中获得有效的网络结构特征,因此表现出来的分类性能之间的差异性就越小.

基于表 2~4 和图 2 的结果,本文得到 4 点结论:

1) 在 Citeseer 稀疏网络上,基于高阶表示的 GraRep 其网络节点分类性能不如 DeepWalk 优异,但是在 DBLP 和 SDBLP 等稠密网络数据集上, GraRep 的网络节点分类性能优于 DeepWalk.该结果表明,在稀疏网络上,高阶的网络表示学习算法倾向于获得较差的分类效果.主要是由于网络稀疏,通过矩阵多次相乘获得的高阶特征无法准确地反映出网络结构特征.

2) 在 DBLP 和 SDBLP 等稠密网络数据集上,基于矩阵分解的网络表示学习算法比基于浅层神经模型的网络表示学习算法更有效,前者在网络节点分类任务中性能稍优于后者.但由于矩阵分解效率和复杂度的限制,基于矩阵分解的网络表示学习算法在大规模网络表示学习任务中不可行.主要的原因是,在稠密网络中,网络节点之间的连边数较多,基于纯结构的随机游走算法能够充分地挖掘出节点所蕴含的网络结构,因此不同网络表示学习算法之间的差异性不明显.假如引入的外部特征噪音较多,反而会使使得网络表示学习的性能受到影响.

3) 网络节点的文本特征融入方式较多.本文实验对比了拼接方法、诱导矩阵补全和类知识表示的节点三元组约束等方法.其中,简单的向量拼接方法并不能带来网络表示性能大幅度的提升.诱导矩阵补全是一种非常有效的文本特征融入框架,在 3 类真实数据集上均获得了优异的网络表示性能.本文提出的文本特征融入框架克服了矩阵分解的计算限制,采用节点三元组对 DeepWalk 训练过程进行约束,从而使得到的网络表示中含有语义信息,也使得网络表示含有更多的信息.

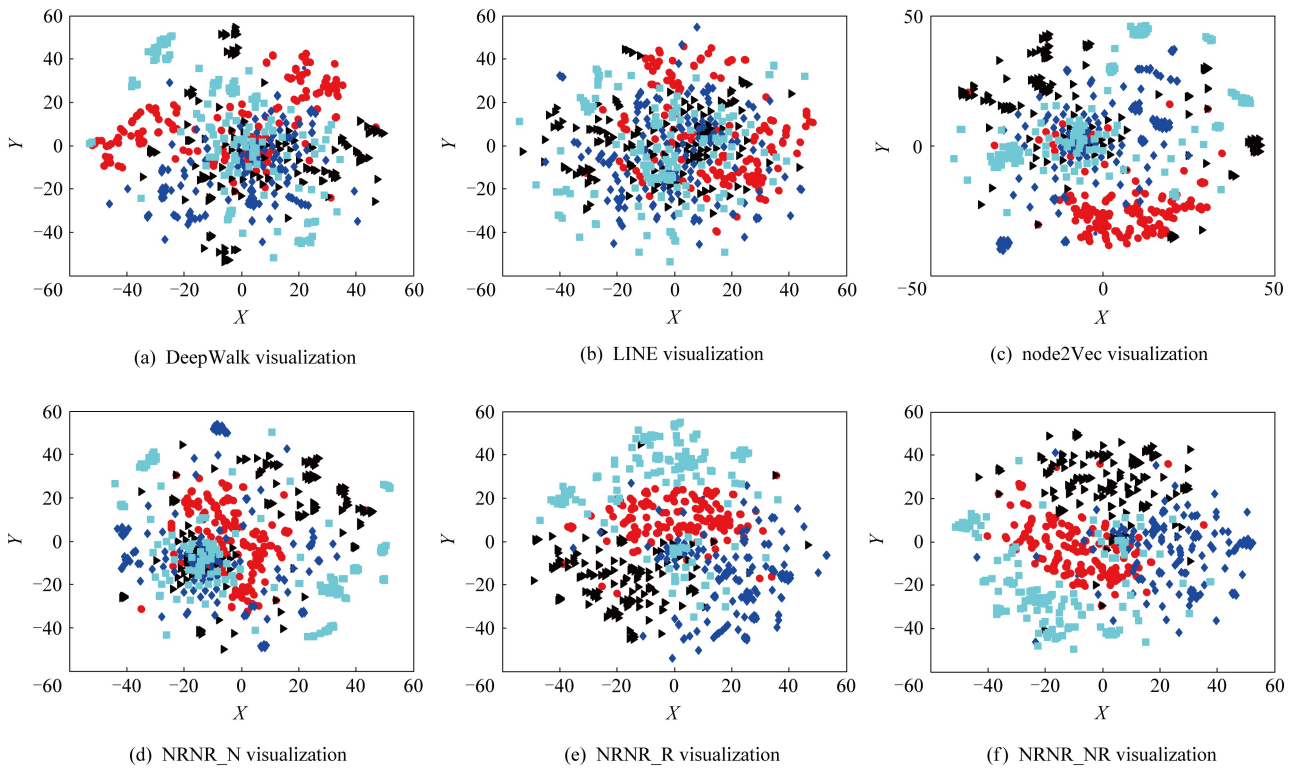
4) 在 Citeseer 等稀疏网络上,通过优化随机游走过程中一阶邻居节点, NRNR_N 的分类性能优于 DeepWalk,但是随着网络节点的平均度的增大,这种差异会变得越来越小.例如在 DBLP 和 SDBLP 数据集上, NRNR_N 和 DeepWalk 之间的分类性能差异变得很小.通过将 2 种优异的网络表示学习算法融合学习,得到的算法在网络节点分类任务中其性能优于其中的一种算法,例如本文使用 NRNR_NR

融合了 NRNR_N 和 NRNR_R 这 2 种算法的改过程,其性能在 3 类数据集上均获得了优于其中一种算法的效果.

3.5 网络表示可视化

网络表示可视化的主要目的是查看训练得到的表示向量能否出现明显的聚类现象,如果出现了聚类现象,则在网络节点分类和链路预测等任务中可

发挥出优异的性能.聚类现象可被认为是网络表示是否学习得到了网络的社团信息.基于网络表示得到的社团划分越准确,则在网络节点分类任务中具有更好的可靠性.在此实验中,本文从 Citeseer 数据集中随机选取了 4 类节点,每个类别随机选取 150 个节点.本文使用 t-SNE 算法可视化学习得到的网络表示.具体结果如图 3 所示:



Different shapes and colors represent different types of vertices

Fig. 3 The network visualization results of the partial vertices on the Citeseer dataset

图 3 网络部分节点表示在 Citeseer 数据集上可视化结果

从图 3 可以发现,本文可视化了 DeepWalk, LINE, node2Vec 与本文提出的 3 类模型训练得到的部分网络表示向量.从表 2~4 可知,DeepWalk 和 LINE 的网络表示分类性能较差,因此,这 2 种算法学习得到的网络表示向量在可视化任务中也同样表现出了较差的结果. node2Vec 和 NRNR_N 可视化结果中,仅有 2 类节点表现出凌乱的分布,其余 2 类颜色的节点具有明显的聚类边界.本文提出的 NRNR_R 和 NRNR_NR 的可视化结果中,4 类节点的表示向量均展现出了明显的聚类现象和聚类边界.该可视化实验说明,在网络表示学习任务中,网络 1 阶邻居节点优化、网络节点文本优化或者 2 种优化联合学习的策略均能提高网络表示学习的性能.

3.6 案例研究

在该案例研究中,首先使用“An Evolutionary Algorithm That Constructs Recurrent Neural Networks”作为目标节点的文本,然后返回与该目标节点最相似的 3 个节点.本实验通过余弦相似度计算节点之间的相似度值.

如表 5 所示,DeepWalk 算法基于网络的结构特征学习网络表示,未考虑节点的文本相似性,因此返回的相似节点中最相关的节点文本中不含有目标节点文本中的词语. TADW 和 NRNR_NR 均考虑了节点的文本特征和结构特征,因此,返回的相似节点中,其节点文本与目标节点的文本之间均有相同的词语共现.由于不同的算法挖掘网络节点之间的

Table 5 Case Study on Citeseer

表 5 Citeseer 数据集上的案例分析

Algorithm	Vertex Title	Similarity Degree
DeepWalk	The Evolution of Communication in Adaptive Agents	0.827 4
	Challenges and Opportunities of Evolutionary Robotics	0.803 1
	Applying Evolutionary Computation to Design Neural Networks: A Study of the State of the Art	0.799 8
TADW	Pareto Evolutionary Neural Networks	0.770 5
	Compensating for Neural Transmission Delay using Extrapolatory Neural Activation in Evolutionary Neural Networks	0.749 1
	An Adaptive Merging and Growing Algorithm for Designing Artificial Neural Networks	0.744 9
NRNR_NR	Applying Evolutionary Computation to Design Neural Networks: A Study of the State of the Art	0.823 5
	Evolutionary Algorithms for Neural Network Design and Training	0.807 6
	Making Use of Population Information in Evolutionary Artificial Neural Networks	0.802 9

不同结构特征,因此返回的最相似节点的文本可能并不相同。

行性.在未来研究中将研究如何增大不同社区或类别的节点在向量表示空间中的距离。

4总 结

参 考 文 献

本文首次引入了邻节点优化策略提升网络表示学习性能,在窗口大小为 5 时,当前节点的 1 阶邻节点被优化,自然地与 2 阶邻节点产生了不同,进而导致一阶邻节点和 2 阶邻节点对当前中心节点的影响不同.该 1 阶邻节点的优化过程被认为是为窗口内的上下文节点赋予不同的位置信息.本文首次引入了关系模型进入网络表示学习,该方法是一种新颖的文本嵌入网络表示的方法.不同与 TADW 和 CENE 等单独训练文本特征的方式,本文提出的 NRNR 方法在训练网络表示学习模型的同时建模了节点与文本之间的关联关系.为了实现该过程,本文进而引入了节点关系三元组概念,使得节点之间含有共同词语的节点对在向量表示空间中具有更近的网络表示.本文提出了 3 个目标函数,分别讨论了 1 阶邻节点、节点三元组与其 2 者联合学习对网络表示学习性能的影响.实验结果表明,添加了 1 阶优化的网络表示学习 NRNR_N 算法在网络节点分类任务中其性能优于原始的 DeepWalk 算法.使用关系模型建模节点与文本内容的 NRNR_R 算法在网络节点分类任务中其性能优于 TADW 算法.如果将 NRNR_N 和 NRNR_R 相结合学习,则得到的 NRNR_NR 算法在网络节点分类任务中其性能均优于本文提出的各类对比算法.在网络可视化测试中,本文提出的 NRNR_R 和 NRNR_NR 均展示出了明显的聚类现象,其聚类边界明显.因此,本文所做的实验地验证了本文提出的 NRNR 算法具有可

[1] Fernando G, Gapar M, Mdjohar M J. Framework for social network data mining [J]. New York Times Book Review, 2015, 116(18): 7-10

[2] Perozzi B, Ai-rfou R, Skiena S. DeepWalk: Online learning of social representations [C] //Proc of the 20th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2014: 701-710

[3] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality [C] //Proc of the 27th Annual Conf on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2013: 3111-3119

[4] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [EB/OL]. (2013-01-16) [2018-10-15]. <https://arxiv.org/pdf/1301.3781.pdf>

[5] Grover A, Leskovec J. node2Vec: Scalable feature learning for networks [C] //Proc of the 22nd ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. New York: ACM, 2016: 855-864

[6] Perozzi B, Kullami V, Chen Haochen, et al. Don't walk, skip! online learning of multi-scale network embeddings [C] //Proc of the 9th IEEE/ACM Int Conf on Advances in Social Networks Analysis and Mining. Piscataway, NJ: IEEE, 2017: 258-265

[7] Yang Cheng, Liu Zhiyuan, Zhao Deli, et al. Network representation learning with rich text information [C] //Proc of the 24th Int Joint Conf on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2015: 2111-2117

[8] Natarajan N, Dhillon I S. Inductive matrix completion for predicting gene-disease associations [J]. Bioinformatics, 2014, 30(12): 60-68

- [9] Pan Shirui, Wu Jia, Zhu Xingquan, et al. Tri-party deep network representation [C] //Proc of the 25th Int Joint Conf on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2016: 1895-1901
- [10] Sun Xiaofei, Guo Jiang, Ding Xiao, et al. A general framework for content-enhanced network representation learning [EB/OL]. (2016-07-24) [2018-10-13]. <https://arxiv.org/abs/1610.02906>
- [11] Zhu Shenghuo, Yu Kai, Chi Yun, et al. Combining content and link for classification using matrix factorization [C] //Proc of the 30th Annual Int ACM SIGIR Conf on Research and Development in Information Retrieval. New York: ACM, 2007: 487-494
- [12] Bhagat S, Cormode G, Muthukrishnan S. Node classification in social networks [EB/OL]. (2011-03-14) [2018-10-16]. <https://arxiv.org/abs/1101.3291>
- [13] Bollacker K, Evans C, Parithosh P, et al. Freebase: A collaboratively created graph database for structuring human knowledge [C] //Proc of the 2008 ACM SIGMOD Int Conf on Management of Data. New York: ACM, 2008: 1247-1250
- [14] Bordes A, Usunier N, Garcia-duran A. Translating embeddings for modeling multi-relational data [C] //Proc of the 27th Annual Conf on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2013: 2787-2795
- [15] Bengio Y, Schwenk H, Senencl J S, et al. Neural probabilistic language models [J]. Journal of Machine Learning Research, 2003, 3(6): 1137-1155
- [16] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives [J]. IEEE Transactions on PAMI. 2013, 35(8): 1798-1828
- [17] Yi Long, Liu Yuling. Text coverless information hiding based on Word2Vec [J]. Journal of Internet Technology, 2018, 19(3): 649-655
- [18] Zhang Dongwen, Xu Hua, Su Zengcai, et al. Chinese comments sentiment classification based on Word2Vec and SVM [J]. Expert Systems with Applications, 2015, 42(4): 1857-1863
- [19] Li Qiuxia, Liu Dan. Research of music recommendation system based on user behavior analysis and Word2Vec user emotion extraction [C] //Proc of the 2nd Int Conf on Intelligent and Interactive Systems and Applications. Berlin: Springer, 2017: 469-475
- [20] Zhang Rui, Tang Sheng, Liu Wu, et al. Multi-modal tag localization for mobile video search [J]. Multimedia Systems, 2017, 23(6): 713-724
- [21] Lu Linyuan, Zhou Tao. Link prediction in complex networks: A survey [J]. Physica A Statistical Mechanics & Its Applications, 2011, 390(6): 1150-1170
- [22] Gao Sheng, Dennyer L, Gallinari P. Temporal link prediction by integrating content and structure information [C] //Proc of the 20th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2011: 1169-1174
- [23] Liu Zemin, Zheng V W, Zhao Zhou, et al. Distance-aware DAG embedding for proximity search on heterogeneous graphs [C] //Proc of the 32nd AAAI Conf on Artificial Intelligence. Palo Alto: AAAI, 2018: 2355-2362
- [24] Tang Jian, Liu Jingzhou, Zhang Ming, et al. Visualizing large-scale and high-dimensional data [C] //Proc of the 25th Int Conf on World Wide Web. Berlin: Springer, 2016: 287-297
- [25] Tang Jian, Qu Meng, Wang Mingzhe, et al. LINE: Large-scale information network embedding [C] //Proc of the 24th Int Conf on World Wide Web. Berlin: Springer, 2015: 1067-1077
- [26] Yang Cheng, Sun Maosun, Liu Zhiyuan, et al. Fast network embedding enhancement via high order proximity approximation [C] //Proc of the 26th Int Joint Conf on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2017: 3894-3900
- [27] Cao Shaosheng, Lu Wei, Xu Qionghai. GraRep: Learning graph representations with global structural information [C] //Proc of the 24th ACM Int on Conf on Information and Knowledge Management. New York: ACM, 2015: 891-900
- [28] Li Juzheng, Zhu Jun, Zhang Bo. Discriminative deep random walk for network classification [C] //Proc of the 54th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: ACL, 2016: 1004-1013
- [29] Tu Cunchao, Zhang Weicheng, Liu Zhiyuan, et al. Max-margin deepwalk: Discriminative learning of network representation [C] //Proc of the 25th Int Joint Conf on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2016: 3889-3895
- [30] Zhang Xia, Chen Weizheng, Yan Hongfei. TLINE: Scalable transductive network embedding [C] //Proc of Asia Information Retrieval Symp. Berlin: Springer, 2016: 98-110
- [31] Chen Jifan, Zhang Qi, Huang Xuanjing. Incorporate group information to enhance network embedding [C] //Proc of the 25th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2016: 1901-1904
- [32] Li Chaozhao, Li Zhoujun, Wang Senzhang, et al. Semi-supervised network embedding [C] //Proc of Int Conf on Database Systems for Advanced Applications. Berlin: Springer, 2017: 131-147
- [33] Wang Suhang, Tang Jiliang, Aggrwal C C, et al. Linked document embedding for classification [C] //Proc of the 25th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2016: 115-124
- [34] Zhang Daokun, Yin Jie, Zhu Xingquan, et al. Collective classification via discriminative matrix factorization on sparsely labeled networks [C] //Proc of the 25th ACM Int Conf on Information and Knowledge Management. New York: ACM, 2016: 1563-1572

- [35] Yang Zhilin, Cohen W W, Salakutdinov R. Revisiting semi-supervised learning with graph embeddings [C] //Proc of the 33rd Int Conf on Machine Learning. New York: ACM, 2016: 40-48
- [36] Huang Xiao, Li Jundong, Hu Xia. Label informed attributed network embedding [C] //Proc of the 10th ACM Int Conf on Web Search and Data Mining. New York: ACM, 2017: 731-739
- [37] Zhou Lekui, Yang Yang, Ren Xiang, et al. Dynamic network embedding by modeling triadic closure process [C] //Proc of the 32nd AAAI Conf on Artificial Intelligence. Palo Alto: AAAI, 2018: 571-578
- [38] Ma Jianxin, Cui Peng, Zhu Wenwu. DepthLGP: Learning embeddings of out-of-sample nodes in dynamic networks [C] //Proc of the 32nd AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2018: 370-377
- [39] Tu Ke, Cui Peng, Wang Xiao. Structural deep embedding for hyper-networks [C] //Proc of the 32nd AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2018: 426-433
- [40] Goodfellow I J, Pouget-abadie J, Mirza M, et al. Generative adversarial networks [J]. Advances in Neural Information Processing Systems, 2014, 3(13): 2672-2680
- [41] Dai Quanyu, Li Qiang, Tang Jian. Adversarial network embedding [C] //Proc of the 32nd AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2018: 2167-2174
- [42] Wang Hongwei, Wang Jia, Wang Jialin, et al. GraphGAN: Graph representation learning with generative adversarial nets [C] //Proc of the 32nd AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2018: 2508-2515
- [43] Bojchevshi A, Shcuur S, Zugner D, et al. NetGAN: Generating graphs via random walks [C] //Proc of the 35th Int Conf on Machine Learning. New York: ACM, 2018: 609-618
- [44] Liu Zhiyuan, Sun Maosong, Lin Yankai, et al. Knowledge representation learning: A review [J]. Journal of Computer Research and Development, 2016, 53(2): 247-261 (in Chinese)
(刘知远, 陈茂松, 林衍凯, 等. 知识表示学习研究进展[J]. 计算机研究与发展, 2016, 53(2): 247-261)

- [45] Fan Ronggen, Chang Kaiwei, Hsieh C J, et al. LIBLINEAR: A library for large linear classification [J]. Journal of Machine Learning Research, 2008, 9(9): 1871-1874



Ye Zhonglin, born in 1989. PhD, student member of CCF. His main research interests include data mining, knowledge discovery and natural language representation learning and understanding.



Zhao Haixing, born in 1969. PhD, professor, PhD supervisor. He is a director of Changjiang Scholars and Innovative Research Team in University, he is also the syndic of Operations Research Society, Combinatorics and Graph Theory Society in China. Member of CCF. His main research interests include complex network, semantic network and machine translation, hypergraph theory and database, network reliability.



Zhang Ke, born in 1986. PhD, student member of CCF. His main research interests include hypergraph and complex network. (hbsanli@163.com)



Zhu Yu, born in 1986. PhD, student member of CCF. His main research interests include link prediction and network representation learning. (sdcxzhuyu@163.com)



Xiao Yuzhi, born in 1980. PhD, member of CCF. Her main research interests include complex network analysis and mining, multi-layer network construction and modeling, hyper-graph theory. (qh_xiaoyuzhi@139.com)