中国科学: 信息科学 2021 年 第 51 卷 第 7 期: 1116–1130
SCIENTIA SINICA Informationis
论文

Network-splitter: 一种基于重叠社区的网络特征提取算法及其在链路预测中的应用

廖好[1], 黄晓敏[1], 吴子强[1], 周明洋[1*], 毛睿[1], 汪秉宏[2]

1. 深圳大学计算机与软件学院, 深圳 518060
2. 中国科学技术大学近代物理系, 合肥 230027
* 通信作者. E-mail: zmy@szu.edu.cn

收稿日期: 2020–07–13; 修回日期: 2020–09–07; 接受日期: 2020–10–11; 网络出版日期: 2021–06–30

国家自然科学基金 (批准号: 61803266, 61703281, 62072311, 71874172)、广东省自然科学基金 (批准号: 2019A1515011173, 2019A1515011064, 2017B030314073) 和深圳市自然科学基金 (批准号: JCYJ20190808162601658, JCYJ20180305124628810) 资助项目

摘要 链路预测任务是根据已知的网络结构和节点属性等信息来预测网络中产生新链路的可能性. 它是网络科学中的一个基础性问题, 具有重要的理论研究和实际应用价值. 近年来, 网络表示学习领域的学者利用深度学习提取网络复杂特征, 大幅度提高了链路预测效果. 实际网络中节点具有局部聚类现象, 然而, 当前的网络表示学习侧重于提取网络全局特征, 忽略了局部信息特征. 针对这个问题, 我们提出了能够学习网络中节点在不同社区中局部特征表示的模型 network-splitter. 该模型利用重叠社区思想, 在每个社区中创建节点的一个角色副本, 并学习该角色副本的特征表示. 最后将节点在不同社区中对应的角色副本信息通过神经网络综合, 得到的综合向量包含网络全局特征和节点局部特征, 并可应用到链路预测任务中. 本文的实验结果表明, network-splitter 模型与最新的网络学习表示方法相比具有很强的竞争力.

关键词 链路预测, 复杂网络, 网络表示学习, 局部节点特征, 重叠社区

1 引言

网络的形式可以很好地描述社会、生物和信息系统. 预测网络中已经丢失的或是未来可能产生的连边, 是非常有价值的研究方向 [1~5]. 从广义上讲, 链路预测主要解决两种类型的问题: 根据不完整的网络拓扑结构对网络进行虚假连边过滤, 和对未来可能产生的链路进行预测. 前者专注于分析网络结构并收集有关各个节点之间链接的信息, 例如, 社交网络中通过分析网络结构可以解决数据丢失问题 [6], 或识别虚假链接在网络中的传播 [7]. 后者着重于预测网络结构中未来链路的问题 [8], 例如, 在

引用格式: 廖好, 黄晓敏, 吴子强, 等. Network-splitter: 一种基于重叠社区的网络特征提取算法及其在链路预测中的应用. 中国科学: 信息科学, 2021, 51: 1116–1130, doi: 10.1360/SSI-2020-0209
Liao H, Huang X M, Wu Z Q, et al. Network-splitter: a network feature extraction algorithm based on overlapping community and its application in link prediction (in Chinese). Sci Sin Inform, 2021, 51: 1116–1130, doi: 10.1360/SSI-2020-0209

社交网络中通过推断潜在的朋友关系来为用户推荐新朋友, 提高用户在在线社交和交互发现方面的体验 [9,10]; 在生物网络中, 通过识别蛋白质网络上的潜在链接, 节省大量的人工盲目检查工作, 大大降低实验成本. 因此, 链路预测引起了研究者的广泛关注 [11,12].

传统的链路预测方法可以分为几类: 基于局部链路预测方法是在假设两个节点有多个公共邻域的情况下进行预测 [13,14], 这类基于局部相似性的方法只考虑了局部网络结构, 在稀疏网络或网络规模较大的情况下预测精度较低; 全局链路预测方法考虑了整个网络的结构相似性 [15,16], 这类方法具有更高的链路预测精度, 但是它们的计算成本也随之升高, 使得算法复杂度过高而不适用于大规模网络; 基于概率统计的链路预测方法假设网络存在已知的先验结构 [17,18], 这类概率模型通常使用大量参数来建立, 由条件概率确定给定节点对之间的链接是否存在.

网络表示学习是近几年提出的解决网络分析问题的高效方法, 它将网络形式的数据映射到低维空间中, 让网络的结构信息和节点属性得到最大限度的保留. 近年来, 基于网络表示学习模型 [19,20] 的链路预测算法层出不穷. 这些算法的思想是学习一个映射函数, 将丰富的网络结构信息以向量的形式嵌入到潜在空间中, 根据节点向量的距离来度量节点对的相似性. 目前大部分基于网络表示学习的链路预测方法的关键是为网络中的每个节点学习单个嵌入向量 [21]. 而在节点非重叠聚类方法的研究中发现, 每个节点至少会属于一个集群或社区 [22,23].

传统的方法保留了网络中的全局信息, 但很容易掩盖其中的局部信息, 群落结构就是其中重要的局部信息 [24]. 当网络具有重叠群落结构时, 这个特征在经典的网络表示中无法体现. 这一问题引发了我们的思考: 我们能否提出一种节点嵌入方法, 让节点的特征向量能表示它们在不同社区中的信息? 本文针对当前研究现状, 综合网络表示学习和节点非重叠聚类方法, 将对节点的嵌入研究扩展到学习节点在多社区中的特征向量表示. 在此基础上使用神经网络, 将不同社区中的向量表示成一个综合的节点向量, 该综合向量将包含节点在不同社区中的信息和属性. 本文在 4 个真实网络中进行了实验, 结果表明本文提出的算法能够提高链路预测的精确度.

2 相关工作

2.1 传统的链路预测方法

传统的链路预测方法主要是基于网络的局部结构和节点属性来计算节点间的相似性, 两个节点之间相似性越大, 它们之间存在连边的可能性就越大. 本文中将 $N(u)$ 表示为节点 u 的邻居节点集合, 它的值是节点 u 邻居节点的个数, $g(u,v)$ 表示节点 u 和 v 的相似性得分.

(1) Adamic 和 Adar 共同提出了 AA 算法 [9], 该算法的基本思想是: 节点 u 的不同邻居节点, 相似性的权重是不同的. 例如, 两个同时喜欢一首冷门小众歌曲的人, 比两个同时喜欢一首热门的流行歌曲的人, 更能说明他们有相似的兴趣品味. 节点 u 和 v 的相似性表示如下:

$$g(u,v)=\sum_{x\in N(u)\bigcap N(v)}\frac{1}{\log(|N(x)|)}. \tag{1}$$

(2) RA 算法 [13] 将 AA 算法中的 $\log(|N(x)|)$ 改为节点 x 的度 $|N(x)|$, 当网络中节点的平均度较大时, $\log(|N(x)|)$ 和 $|N(x)|$ 的值就有很大区别, 两个算法的预测能力也大不相同.

$$g(u,v)=\sum_{x\in N(u)\bigcap N(v)}\frac{1}{|N(x)|}. \tag{2}$$

(3) Jaccard 算法 [25] 的定义方式是节点 u 和 v 的共同邻居数与该节点对所有邻居数之比. 若节点对共同邻居的数量越多, 且所有邻居节点越少, 那么该节点对的相似性得分 $g(u,v)$ 越高.

$$g(u,v)=\frac{|N(u)\bigcap N(v)|}{|N(u)\bigcup N(v)|}. \tag{3}$$

2.2 基于网络表示学习的链路预测方法

(1) Perozzi 等 [26] 提出的 Deepwalk 算法由两部分组成: 第 1 部分是随机游走算法, 第 2 部分是用 Word2vec 模型 [27] 将第 1 部分得到的游走序列转化成节点的表示向量. 网络 $G=(V,E)$, 其中 V 表示图中的节点, E 表示图中的连边. 对于第 1 部分随机游走算法, 从图 G 中的任一节点 $v_i(v_i\in V)$ 出发, 采用深度优先搜索的方式, 可以随机选择游走至当前节点的任一邻居节点, 并将该节点添加至随机游走序列 W_i 中, 直到 W_i 的长度等于设置的值 l, 则节点 v_i 的游走结束. Deepwalk 算法将由节点组成的随机游走序列看作由词语组成的句子. Word2vec 模型将网络 G 中的每个节点 v_i 映射成一个高维向量, 越相似的节点将被映射成距离越近的向量, 通过对向量的运算实现节点间的链路预测.

(2) Node2vec 算法是对 Deepwalk 算法的改进 [28]: 它提出了偏随机游走的方法, 改进了 Deepwalk 算法基于深度优先搜索的随机游走策略. 该算法定义了两个参数: 返回概率参数 p 和离开概率参数 q. 这两个参数让游走过程不再只是用深度优先搜索的方式, 而是结合了广度优先搜索和深度优先搜索的一种有偏的随机游走, 从而得到更准确的节点表示. 具体来说, 通过偏随机游走过程得到游走序列 W_i, 再使用 Word2vec 模型将网络 G 中的每个节点 v_i 映射成一个高维向量.

(3) LINE 方法将包含数百万个节点的大规模信息网络嵌入到低维向量空间, 能够适用于无向或有向、加权或无权的信息网络 [29]. LINE 模型定义了两种节点相似性. 其中, 一阶相似性是网络中两个相连节点对的相似度, 二阶相似性是节点对邻居结构的相似性. 同时该方法优化了目标函数, 解决了经典随机梯度下降算法的局限性.

(4) Graph-GAN 算法将对抗生成网络的思想加入到了网络表示学习的研究中 [30]. 该模型由生成模型和判别模型组成: 生成模型拟合真实连边的正样本, 并生成一些假连边的负样本来欺骗判别模型; 判别模型负责判断两个节点之间的连边关系是真实的还是由生成模型生成的, 它的目标是最大化区分正样本和负样本. 两种模型不断相互竞争从而交替迭代提升算法的精确度.

(5) 标准的网络表示学习方法是为每个节点学习单个向量来表示该节点的特征. 真实网络结构中的节点可能属于多个重叠的社区, 即同一个节点在不同的社区扮演着不同的角色. 基于这个观点, 谷歌团队在 WWW 2019 提出的 splitter 算法是一种无监督的嵌入方法, 该方法为网络中的每个节点嵌入多个向量, 以便更好地表示节点在一些重叠社区中的特征 [31].

2.3 链路预测常用的评价指标

(1) **AUC (area under ROC curve).** 在链路预测任务中, AUC 可以理解为在测试集中随机选取一条连边, 比随机选取一条不存在的连边得分高的概率 [32]. 在实验中, 每次从测试集的连边中随机选取一条, 再从不存在的连边中随机选取一条, 如果测试集中选取的连边得分高, 就加 1 分; 如果两条连边得分相等, 就加 0.5 分; 如果测试集中选取的连边得分低, 就加 0 分. 重复 n 次实验, 如果有 n' 次测试集中的连边得分高, 有 n'' 次两条连边得分相等, 那么 AUC 可以定义为:

$$\mathrm{AUC}=\frac{n'+0.5n''}{n}. \tag{4}$$

(2) **Precision 和 Accuracy.** 在链路预测任务中, 随机从测试集的连边和不存在的连边中选取一条, 如果测试集的连边被预测为正样本时 TP 加 1, 被预测为负样本时 FP 加 1; 如果不存在的连边被

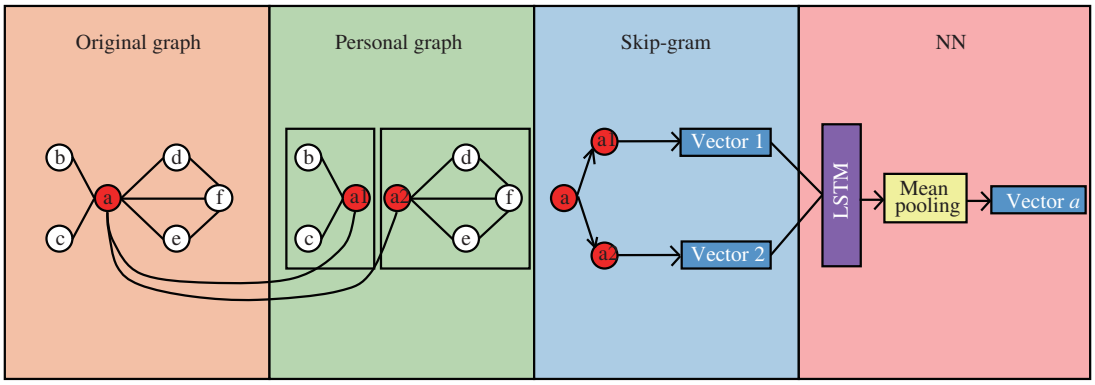


图 1 (网络版彩图) Network-splitter 模型的具体构建过程

Figure 1 (Color online) The modeling process of network-splitter

预测为正样本时 FN 加 1, 被预测为负样本时 TN 加 1. 重复多次实验, 那么 Precision 和 Accuracy 可以分别定义为:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{5}$$

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{6}$$

3 研究目的

文献 [31] 中指出单个向量不足以最好地描述节点属性, 所以在这项工作中, 作者实现了一种学习网络中节点多个向量表示形式的方法, 并且将该方法运用于网络中节点的链路预测. 但是该方法在对节点进行链路预测, 计算节点间相似性时存在一个问题: 该算法将一个节点表示成多个高维向量, 每个向量代表节点在某个社区的角色, 在通过向量距离计算节点间相似性时, 只有相似性最大的那个角色向量会被使用, 而其他社区的角色将会被忽略不计. 这将节点映射成多个角色向量的意义就不存在了. 这个问题引发了我们的思考, 能否使用一个包含该节点所有角色信息的综合向量来表示这个节点呢? 本文提出使用神经网络将节点在所有社区的角色向量汇总成一个综合向量. 由于综合向量包含了节点在不同社区的角色信息, 能够解决 splitter 方法在计算节点间的相似性时只保留最相似的某个角色向量的问题. 我们使用了 4 种神经网络模型, 包括 LSTM (long short-term memory) [33], BiLSTM (bidirectional-LSTM) [34], GRU(gated recurrent unit) [35] 和 BiGRU (bidirectional-GRU), 并且将使用了 4 种模型的方法统称为 network-splitter.

4 Network-splitter 模型

Network-splitter 模型主要分成 3 个部分: 第 1 部分是创建原始网络图的角色图 (personal graph), 将原始图中的节点根据所在的不同社区创建多个角色节点; 第 2 部分是使用 skip-gram 模型将节点的多个角色节点映射成高维向量; 第 3 部分是分别使用 LSTM, BiLSTM, GRU 和 BiGRU 这 4 个神经网络, 将节点的多个角色高维向量训练成一个综合的高维向量, 该综合向量包含了该节点在不同角色中的信息. 以使用 LSTM 神经网络为例, 图 1 展示了 network-splitter 模型的具体构建过程.

4.1 角色图

在创建角色图的步骤中, 对于每个节点 u, 分析其自我中心网络, 使用基于标签传播的非重叠社区发现算法 (label propagation algorithm) 将节点 u 的邻居划分为多个社区 [36]. 在每个社区中, 都会创建一个节点 u 的副本, 并将原始图中节点之间的连边映射到角色图中. 在创建角色图时, 为原始节点在不同的社区创建了不同的角色节点, 但是角色图中不会创建新的关系连边, 连边的数量与原始图保持不变. 在图 1 中, 通过分析节点 a 的自我中心网络, 可发现该节点的邻居可划分为两个不同的社区, 因此构造节点 a 的角色图时, 可使用角色副本 $a1$ 和 $a2$ 代替节点 a. 为原始图中的每个节点都创建相应的角色图, 原始的网络图 $G=(V,E)$ 可转化为角色图 $P_G=(P_V,E)$.

4.2 Skip-gram 模型

第 2 部分是使用 Word2vec 中的 skip-gram 模型将每个角色节点映射成高维向量. 首先从角色图 $P_G=(P_V,E)$ 中的任一节点 v_i ($v_i\in P_V$) 出发, 采用随机游走 (random walk) 的方式到达当前节点的某一个邻居节点, 并将该节点添加至随机游走序列 W_i 中, 直到 W_i 的长度等于游走长度 l, 则以节点 v_i 为根节点的游走结束. 对于网络中的每一个节点 v_i, 都会生成长度为 l 的游走序列 W_i.

Skip-gram 模型常用于自然语言处理中, 利用句子中的关键词预测上下文的单词. 在本文中, 可以将由节点构成的序列看作由单词构成的句子, 按同样的方式来预测相邻的节点. 在网络图中, $W_0=(v_0,v_1,v_2,\ldots,v_l)$ 是一个从节点 v_0 出发、由若干个节点组成的游走序列. 在整个训练集上需要优化的目标是

$$P_r(v_l|v_0,v_1,v_2,\ldots,v_{l-1}). \tag{7}$$

它表示当知道 $v_0,v_1,v_2,\ldots,v_{l-1}$ 这 l 个节点的游走序列后, 下一个节点是 v_l 的概率为多少. 由于 v_i 表示节点的 id 无法计算概率, 所以引入映射函数 $\Phi: v\in P_V\to\mathbb{R}^{|P_V|\times d}$, 将所有节点映射为一个 d 维的向量. 此时, 式 (7) 的优化目标可以写成

$$P_r(v_l|\Phi(v_0),\Phi(v_1),\Phi(v_2),\ldots,\Phi(v_{l-1})). \tag{8}$$

计算在节点 v_i 的 ω 窗口范围内其他节点出现的概率, 其中 ω 窗口表示在节点所在的游走序列中从当前节点开始前后各选取 ω 个节点, 通过最小化映射函数 Φ 的误差, 最终获得所有角色节点对应的向量.

$$\min_{\Phi}-\log P_r(\{v_{i-\omega},\ldots,v_{i-1},v_{i+1},\ldots,v_{i+\omega}\}|\Phi(v_i)). \tag{9}$$

4.3 神经网络模型

模型的第 3 部分是将第 2 部分获得的角色向量分别输入不同的神经网络中, 得到的综合向量将包含所有角色向量的信息. 本文使用了 4 种神经网络: LSTM, BiLSTM, GRU 和 BiGRU, 这些是目前最常见的神经网络模型, 实现简单而且原理清晰. 以 LSTM 为例, 它是一种特殊的循环神经网络 (recurrent neural network, RNN), 图 2 展示了该模型的输入门、输出门和遗忘门机制. LSTM 模型可表示为如下公式:

$$f_t=\sigma(W_f\cdot[h_{t-1},x_t]+b_f), \tag{10}$$

$$i_t=\sigma(W_i\cdot[h_{t-1},x_t]+b_i), \tag{11}$$

$$\tilde{C}_t=\tanh(W_C\cdot[h_{t-1},x_t]+b_C), \tag{12}$$

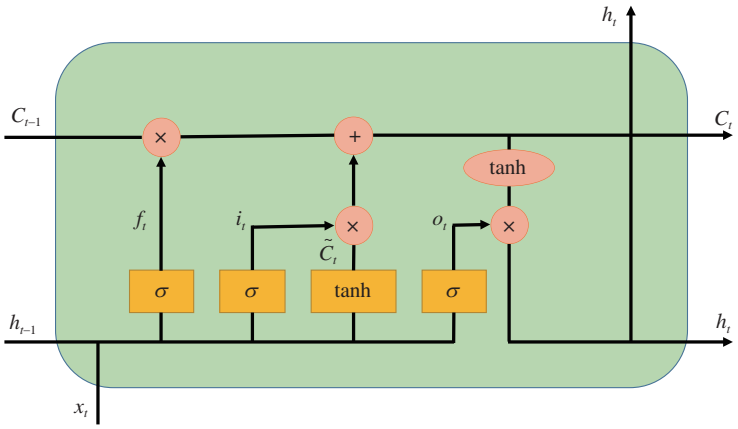


图 2 (网络版彩图) LSTM 模型

Figure 2 (Color online) LSTM model

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t, \tag{13}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{14}$$

$$h_t = o_t * \tanh(C_t), \tag{15}$$

其中, 式 (10) 是遗忘门, 用于计算有多少信息会从 “细胞” 中被丢弃; 式 (11) 和 (12) 为输入门, 它决定了当前有多少有用信息; 式 (13) 用于更新当前 “细胞” 的信息; 输出门由式 (14) 和 (15) 组成, 决定 “细胞” 中哪些信息将作为当前状态被输出. W 表示各个门的权重, b 表示各个门的偏置值, C_t 表示当前 “细胞” 状态, h_{t-1} 表示 $t-1$ 时刻 “细胞” 的输出, x_t 表示 t 时刻 “细胞” 的输入.

GRU 比 LSTM 网络的结构更加简单, 该模型将 LSTM 神经网络中的遗忘门和输入门合并为更新门, 用于计算前一时刻的信息有多少能保留至当前状态. 重置门用于控制前一时刻的信息有多少会被丢弃. 在单向的 LSTM 和 GRU 神经网络结构中, 状态的变化总是从前往后输出的. 将前向 LSTM_l 与后向 LSTM_r 的状态结果相结合就构成了双向的 BiLSTM 神经网络模型, 同样地, 前向的 GRU_l 与后向的 GRU_r 结合成双向的 BiGRU 神经网络模型. 本文使用的是简单且常见的神经网络模型, 扩展到复杂的神经网络模型同样是可行的.

4.4 时间复杂度

Network-splitter 模型在 splitter 的基础上, 结合了神经网络, 将节点的多个角色高维向量训练成一个综合的高维向量. Network-splitter 中使用 LSTM, BiLSTM, GRU 和 BiGRU 4 种神经网络模型时, 分别将算法称为 network-LSTM, network-BiLSTM, network-GRU 和 network-BiGRU. 在算法的时间复杂度上, network-splitter 增加了神经网络的计算, 但是整体而言, 与 splitter 相比, 时间复杂度增加不是非常明显, 以 PPI 数据集为例, 在将节点信息映射为 64 维向量的情况下, 运行时间对比见表 1.

在表 1 中, Network-BiLSTM 的运行时间是 4 个神经网络模型中最长的. Network-splitter 的时间复杂度可以表示为 $O(n^{3/2} + \sqrt{n}T_P(n) + nml\omega(d + d\log(n)) + T_N(d))$. 其中 n 表示角色图中边的数量, $T_P(n)$ 表示利用社区划分算法划分一个有 n 条边的角色图需要花费的时间, m 表示每个节点随机游走的次数, l 表示随机游走的长度, ω 表示窗口的大小, d 表示维度, $T_N(d)$ 表示复杂度最大的神经网络训练 d 维向量需要花费的时间.

表 1 PPI 数据集中 64 维节点向量情况下 network-splitter 模型的运行时间

Table 1 Running time of network-splitter model in 64-dimension-vectorizing PPI dataset

Model	Running time (s)
Splitter	399.48(±44.28)
Network-LSTM	404.04(±44.13)
Network-BiLSTM	410.87(±43.74)
Network-GRU	403.80(±44.19)
Network-BiGRU	409.27(±43.56)

表 2 4 个真实网络的基本属性

Table 2 Basic properties of the four real networks

	Musae-Facebook	Musae-Github	PPI	Power-grid
Nodes	22470	37700	3852	4941
Edges	171002	289003	20881	6594
Publication year	2017	2019	2008	1998
Average degree	15.21	15.33	9.85	2.67

5 实验设置

实验中使用的 4 个真实网络数据集, 具体属性 (包括节点数、连边数、数据集公开年份, 以及平均节点度) 如表 2 所示. 下面是对这 4 个真实网络的详细介绍:

(1) Musae-Facebook[1)]. 该网络是 Facebook 的大型网页网络, 在 2017 年 11 月通过 Facebook graph API 收集. 节点表示 Facebook 上的官方网页, 连边表示网页之间存在跳转链接. 该网络属于社会网络, 网页偏向于政府机构、电视节目等.

(2) Musae-Github[2)]. 该网络是 Github 开发人员的大型社交网络, 于 2019 年 6 月从公共 API 收集. 节点表示至少拥有 10 个存储库的开发人员, 连边表示开发人员之间相互关注的关系. 该网络属于社交网络, 用户偏向于开发人员.

(3) PPI[3)]. 该网络是蛋白质交互网络的子图, 于 2008 年公布. 节点表示蛋白质, 连边表示蛋白质之间有相互作用的关系. 该网络属于生物网络.

(4) Power-grid[4)]. 该网络表示的是美国西部各州电网的拓扑结构, 于 1998 年公布. 节点表示变压器、变电站和发电机, 连边表示高压传输线. 该网络属于电力网络.

本文的实验环境: 处理器 Intel Xeon Silver 4114 CPU 2.20 GHz, 内存 128 GB, 操作系统 CentOS 7.5. 本文中所有实验结果都经过 10 次实验取平均值, 算法的鲁棒性通过标准差来表示. 实验中我们使用了 AUC, Precision 和 Accuracy 作为链路预测的评价指标.

在链路预测任务中, 网络节点的连边被划分为两部分, 分别作为算法的训练集和测试集. 我们将训练集和测试集按照 9:1, 8:2, 7:3, 6:4 和 5:5 进行划分, 网络中节点连边的划分是随机的. 在计算链路预测的精确度时, 不存在的连边的选取也是随机的.

1) http://snap.stanford.edu/data/facebook-large-page-page-network.html.
2) http://snap.stanford.edu/data/github-social.html.
3) https://github.com/Complex-data/Network-Splitter/tree/main/Data/PPI.
4) https://github.com/Complex-data/Network-Splitter/tree/main/Data/Power-grid.

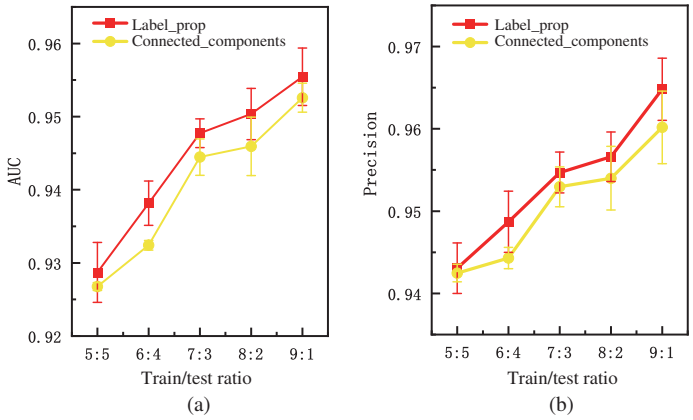


图 3 (网络版彩图) PPI 数据集中 64 维节点向量情况下两种社区划分算法对 network-splitter 的影响

Figure 3 (Color online) Influence of two community partition algorithms on network-splitter in 64-dimension-vectorizing PPI dataset

在 splitter 和 network-splitter 算法中, 第 1 步是使用社区划分算法创建角色图. 我们对比了基于标签传播的非重叠社区发现算法 (label_prop) 和标准的连通分量算法 (connected_components) 的划分结果对算法精确度的影响. 我们将 PPI 数据集分别按照 5:5 至 9:1 的比例划分为训练集和测试集, 对比了在上述两个社区划分算法下, network-splitter 的 AUC 和 Precision, 如图 3 所示. 经过对比, 这两种社区划分算法的实验结果非常接近, 在接下来的实验中, 我们选用了略优的基于标签传播的非重叠社区发现算法用于划分角色图.

6 实验结果

6.1 链路预测算法精确度对比

本小节实验对比了传统的链路预测算法与网络表示学习驱动的链路预测算法在 4 个数据集中的精确度. 我们将数据集划分为 90% 的训练集和 10% 的测试集, 使用训练集建立逻辑回归模型, 通过该模型对测试集的数据进行链路预测. 在本节实验中, 我们对比了 network-splitter 算法和其他 8 种算法的 AUC 和 Precision, 并给出了每个算法精确度的标准差. 在传统的链路预测算法中, 实验结果基本没有波动, 因此没有在表中列出标准差. 网络表示学习驱动的算法将节点信息转化为高维向量, 这里我们将展示 64 维向量的情况. 如表 3 和 4 所示, 其中 network-splitter 算法表示的是 LSTM, BiLSTM, GRU 和 BiGRU 4 种模型中表现最佳的算法.

从表 3 和 4 的结果我们发现, 将节点信息映射成 64 维的向量时, 网络表示学习驱动的链路预测算法 AUC 和 Precision 指标都超过了传统的链路预测算法. Network-splitter 算法在 4 个网络中的精确度都是最高的, 表明该方法在链路预测的应用中表现优异. Power-grid 是 4 个网络中最稀疏的, 大部分链路预测方法在这个网络中精确度低于 0.9, 但是 network-splitter 方法的 AUC 达到了 0.903, Precision 达到了 0.926. 从标准差看来, network-splitter 的鲁棒性良好.

我们将节点信息映射成 8, 16, 32, 64 这 4 种不同维度的向量, 其对比结果如图 4 所示, 可以直观地看到各种网络表示学习驱动的链路预测算法的精确度.

表 3　64 维节点向量情况下各个网络中算法的 AUC 对比

Table 3　AUC comparison of algorithms in 64-dimension-vectorizing networks

	Musae-Facebook	Musae-Github	PPI	Power-grid
AA [9]	0.945	0.855	0.913	0.628
RA [13]	0.945	0.856	0.913	0.628
JC [25]	0.944	0.802	0.912	0.628
Deepwalk [26]	0.957(±0.00015)	0.903(±0.00037)	0.926(±0.00044)	0.798(±0.00089)
Node2vec [28]	0.958(±0.00021)	0.902(±0.00014)	0.926(±0.00011)	0.799(±0.00049)
LINE [29]	0.923(±0.00025)	0.854(±0.00024)	0.893(±0.00027)	0.773(±0.00747)
Graph-GAN [30]	0.942(±0.00029)	0.876(±0.00038)	0.936(±0.00041)	0.766(±0.00035)
Splitter [31]	0.958(±0.00112)	0.886(±0.00308)	0.937(±0.00224)	0.849(±0.01043)
Network-splitter	**0.959(±0.00098)**	**0.914(±0.00017)**	**0.955(±0.00392)**	**0.903(±0.00641)**

表 4　64 维节点向量情况下各个网络中算法的 Precision 对比

Table 4　Precision comparison of algorithms in 64-dimension-vectorizing networks

	Musae-Facebook	Musae-Github	PPI	Power-grid
AA [9]	0.945	0.864	0.914	0.628
RA [13]	0.946	0.864	0.913	0.628
JC [25]	0.944	0.727	0.913	0.627
Deepwalk [26]	0.956(±0.00027)	0.918(±0.00033)	0.922(±0.00083)	0.800(±0.00192)
Node2vec [28]	0.958(±0.00012)	0.917(±0.00016)	0.927(±0.00035)	0.801(±0.00057)
LINE [29]	0.931(±0.00022)	0.859(±0.00058)	0.824(±0.00036)	0.838(±0.00087)
Graph-GAN [30]	0.945(±0.00023)	0.880(±0.00058)	0.926(±0.00048)	0.747(±0.00061)
Splitter [31]	0.963(±0.00146)	0.895(±0.00292)	0.934(±0.00282)	0.887(±0.00693)
Network-splitter	**0.966(±0.00117)**	**0.924(±0.00045)**	**0.965(±0.00377)**	**0.926(±0.00485)**

6.2　不同数据集划分对精确度的影响分析

本小节研究训练集和测试集的划分对链路预测算法精确度的影响, 并给出精确度的标准差. 我们将训练集和测试集按照 9:1, 8:2, 7:3, 6:4 和 5:5 的比例进行划分.

我们对比了 network-splitter 中的 4 种神经网络链路预测算法在 3 个数据集中 AUC, Precision 和 Accuracy 的表现. 由于在 Musae-Facebook 数据集中 4 种算法的结果很接近, 因此我们没有展示. 这部分实验将节点的向量维度固定在 64 维, 对比的结果展示在图 5 中.

由图 5 发现, 在 Musae-Github 数据集中, network-BiGRU 算法采用不同比例的训练集和测试集划分时精确度都最高; 在 PPI 数据集中, 当训练集与测试集比例为 5:5 和 7:3 时, network-BiLSTM 和 network-GRU 的精确度非常接近, network-BiGRU 一直表现出最高的精确度; power-grid 是 3 个数据集中最小的, 算法的精确度受训练集与测试集划分的影响较大, 4 个算法的精确度都随着训练集比例的增大有明显升高, 其中, network-BiLSTM 和 network-LSTM 在 power-grid 数据集中的精确度更高. 上述实验发现: 对于平均节点度较大的网络, 比如社会网络和生物网络, 结合神经网络模型 GRU 和 BiGRU 得到的预测结果更好; 而对于平均节点度较小的网络, 比如电力网络, 结合神经网络模型 LSTM 和 BiLSTM 得到的预测结果更好.

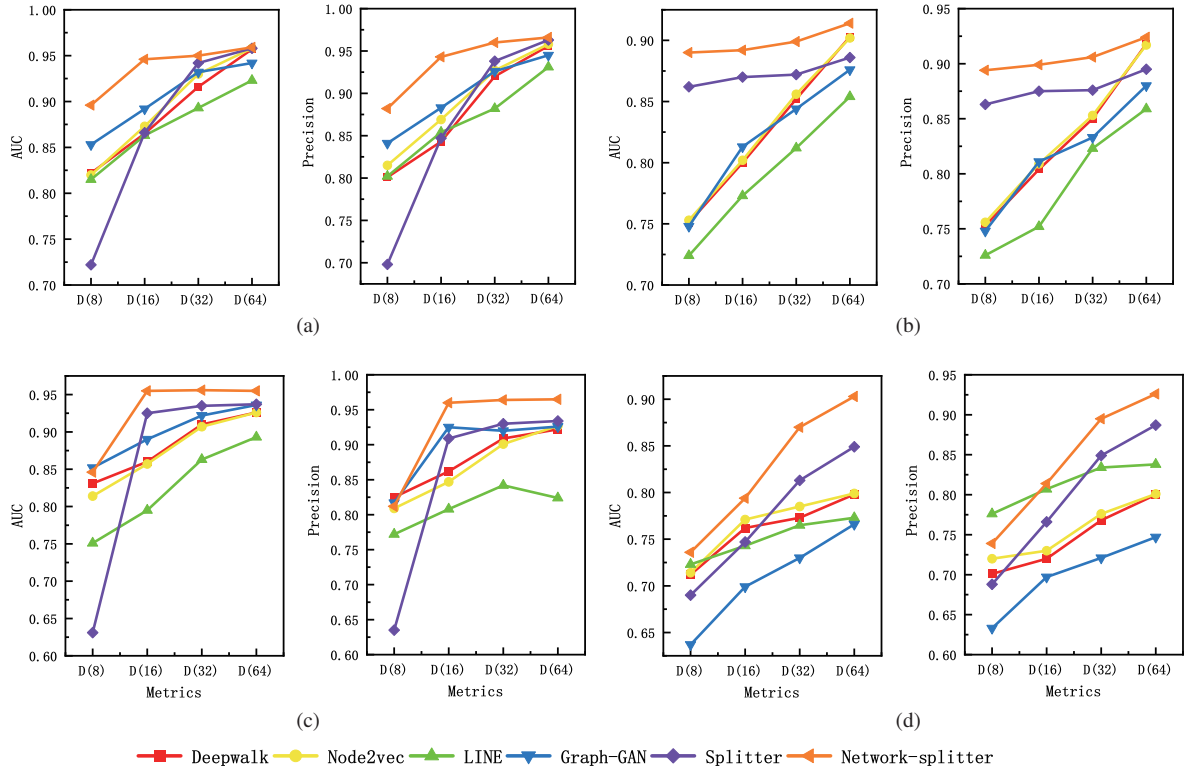


图 4 (网络版彩图) 4 个真实数据集中各维度对算法的影响

Figure 4 (Color online) Influence of different dimensions on the algorithms in the four real datasets. (a) Musae-Facebook; (b) Musae-Github; (c) PPI; (d) power-grid

6.3 节点可视化

这部分实验使用了 Zachary Karate Club (空手道俱乐部) 网络[5], 该网络是网络科学中非常经典的数据, 由 34 个节点和 78 条边组成. 其中 34 个节点可分为 4 类. 该网络的可视化结果, 以及 Node2vec 方法和 network-splitter 方法对节点映射成向量的结果如图 6 所示. 网络表示学习驱动的链路预测算法是通过将节点映射成高维向量后再对节点对进行距离的计算, 向量距离越近的节点越相似. 由于 splitter 方法将节点映射成多个向量后, 不可以直接进行可视化操作, 因此我们仅对比了 splitter 方法与 network-splitter 方法的 AUC. 在该网络中, 节点映射成 64 维向量的情况下, splitter 的 AUC 是 0.76, 而 network-splitter 的 AUC 达到 0.79. 此结果说明 network-splitter 达到了 splitter 的预测精度, 而且可以在可视化结果中直观地展示节点的分类情况.

为了验证算法是否将同一类节点映射到距离更近的空间中, 我们对比了 network-splitter 方法和 Node2vec 方法将节点映射成向量的可视化结果. 使用链路预测方法将节点映射成 64 维向量后进行降维, 在二维的视图中进行节点的可视化实验. 在学习节点属性并通过映射函数将节点映射成高维向量时, 优秀的链路预测方法应将有链接的同类节点映射在更接近的空间中. 在图 6 中, Node2vec 方法学习到的节点信息对不同类别的节点没有做出一个很好的区分, 4 个类别的节点都被映射到了非常接近的空间中. 图 6(c) 和 (d) 中的算法 network-LSTM 和 network-BiLSTM 主要将节点分为了两大类: 红色和蓝色节点在空间中被分为一类, 米色和绿色节点在空间中被分为另一类, 但是两个大类中的不同

5) https://github.com/Complex-data/Network-Splitter/tree/main/Data/Zachary-Karate.

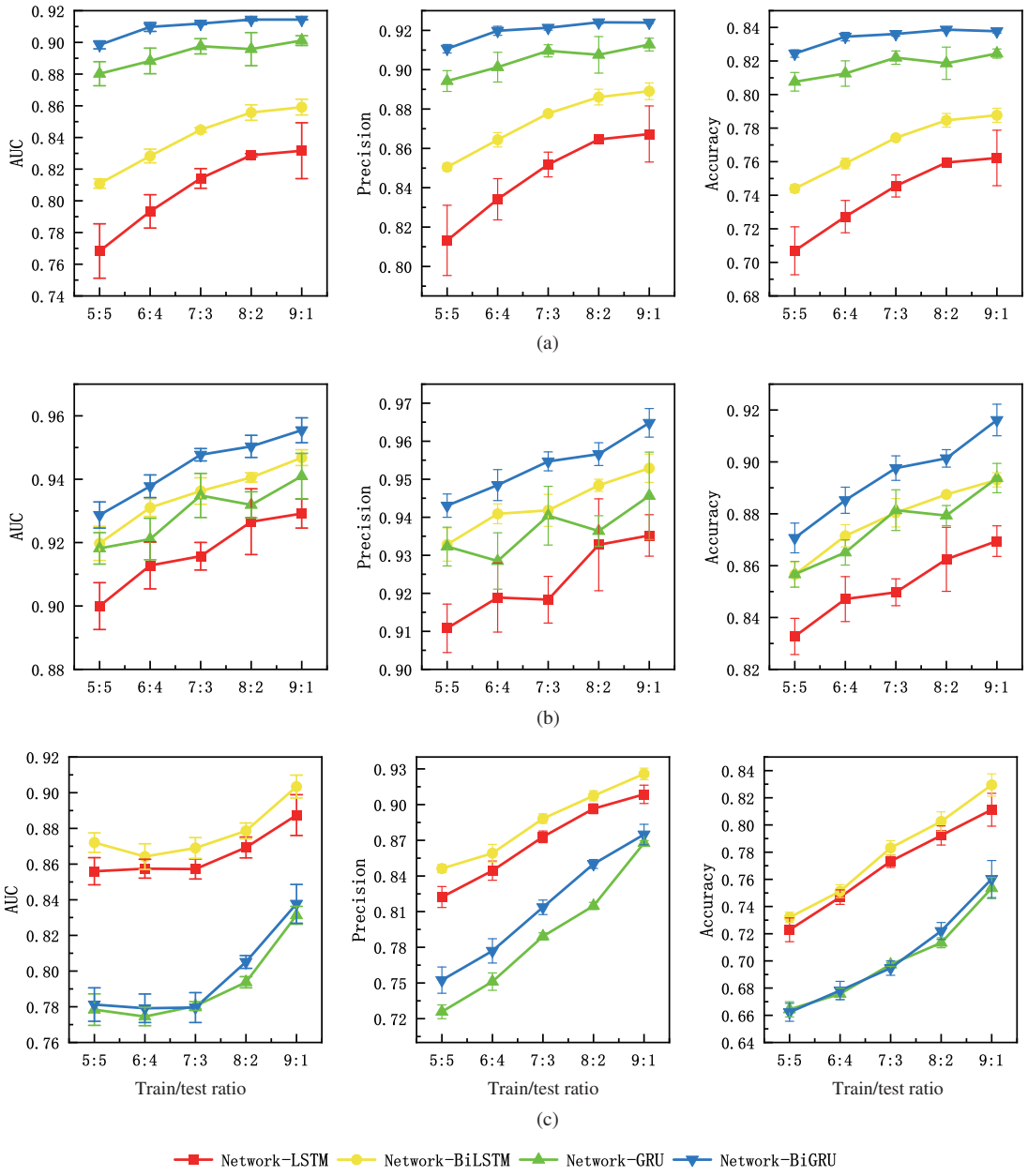


图 5 (网络版彩图) 3 个真实数据集中数据划分对算法的影响

Figure 5 (Color online) Influence of data partition on the algorithms in the three real datasets. (a) Musae-Github; (b) PPI; (c) power-grid

颜色的节点没有被映射到不同的空间中. 这两个算法只学习到了部分节点属性信息. Network-GRU 和 network-BiGRU 对该网络节点信息学习得最充分, 它们都将不同类别的节点在高维空间中做出了很好的分类. 在图 6(a) 中可以看到, 蓝色节点只与红色节点有连边, 与其他两类节点没有连边. Network-GRU 和 network-BiGRU 的方法都学习到了这个属性, 将蓝色节点映射在与红色节点较接近的空间中, 而米色和绿色两类节点都被映射在较远的空间中.

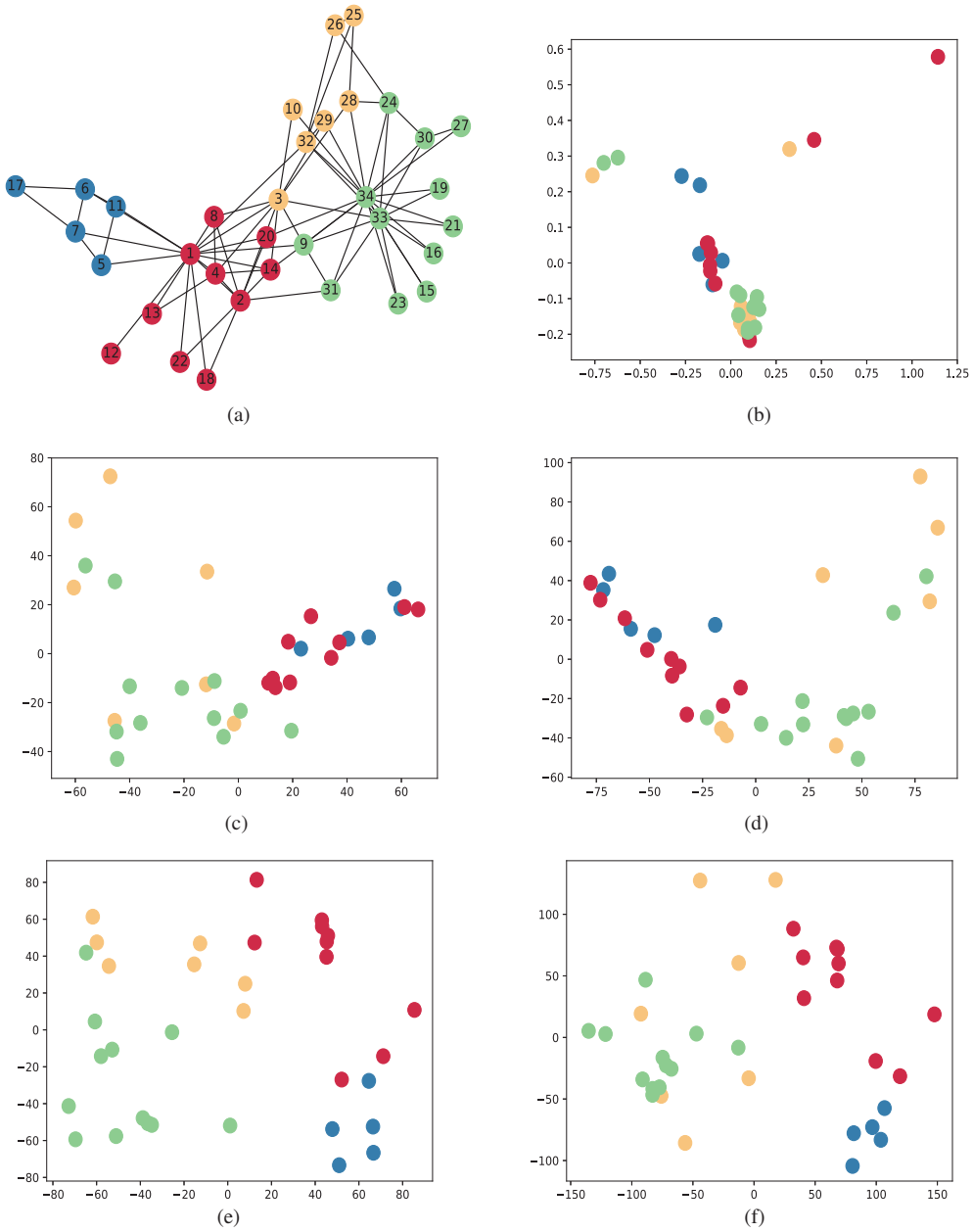


图 6 (网络版彩图) 5 个方法节点可视化结果对比

Figure 6 (Color online) Comparison of visualization results of the five methods. (a) Zachary Karate Club network; (b) Node2vec; (c) network-LSTM; (d) network-BiLSTM; (e) network-GRU; (f) network-BiGRU

7 总结

本文提出了一种基于网络表示学习的链路预测模型 network-splitter. 模型由 3 部分组成, 首先分析目标节点的自我中心网络, 将目标节点的邻居节点进行分类, 生成目标节点的角色图; 然后使用 skip-gram 模型将目标节点在不同社区对应的角色副本节点映射成高维向量; 最后分别利用 4 种神经网络将不同社区的角色副本节点向量训练成一个综合向量, 该综合向量包含目标节点在不同社区的属

性. 本文使用提出的模型进行了 3 个不同的实验, 实验结果均表明, 本文所提出的 network-splitter 模型与现有众多的链路预测算法相比有着更加准确的预测结果. 对于平均节点度较大的网络, 比如社会网络和生物网络, 结合神经网络模型 GRU 和 BiGRU 得到的预测结果更好; 而对于平均节点度较小的网络, 比如电力网络, 结合神经网络模型 LSTM 和 BiLSTM 得到的预测结果更好. 未来工作将研究如何在标签分类或带权网络等其他特定场景中更全面地获取整个网络结构信息, 从而提升模型在不同任务中的精确度. 另外, 这种基于重叠社区的网络特征在时序网络中的节点表示和分类问题也值得探讨.

参考文献

1 Zhang S, Yao L N, Sun A X, et al. Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv, 2019, 52: 1–38
2 Lü L Y, Zhou T. Link prediction in complex networks: a survey. Physica A-Stat Mech Its Appl, 2011, 390: 1150–1170
3 Wang Z T, Chen C Y, Li W J. Predictive network representation learning for link prediction. In: Proceedings of the 40th International ACM SIGIR Conference, 2017. 969–972
4 Costa L D F, Oliveira J O N, Travieso G, et al. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. Adv Phys, 2011, 60: 329–412
5 Wen G H, Yu W W, Yu X H, et al. Complex cyber-physical networks: from cybersecurity to security control. J Syst Sci Complex, 2017, 30: 46–67
6 Kossinets G. Effects of missing data in social networks. Soc Netw, 2006, 28: 247–268
7 Guimerá R, Sales-Pardo M. Missing and spurious interactions and the reconstruction of complex networks. Proc Natl Acad Sci USA, 2009, 106: 22073–22078
8 Dhote Y, Mishra N, Sharma S. Survey and analysis of temporal link prediction in online social networks. In: Proceedings of International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013. 1178–1183
9 Adamic L A, Adar E. Friends and neighbors on the Web. Soc Netw, 2003, 25: 211–230
10 Newman M E J. From the cover: the structure of scientific collaboration networks. Proc Natl Acad Sci USA, 2001, 98: 404–409
11 Watts D J, Strogatz S H. Collective dynamics of 'small-world' networks. Nature, 1998, 393: 440–442
12 Sun J C, Feng L, Xie J R, et al. Revealing the predictability of intrinsic structure in complex networks. Nat Commun, 2020, 11: 574
13 Zhou T, Lü L Y, Zhang Y C. Predicting missing links via local information. Eur Phys J B, 2009, 71: 623–630
14 Barabási A L, Albert R. Emergence of scaling in random networks. Science, 1999, 286: 509–512
15 Rücker G. Network meta-analysis, electrical networks and graph theory. Res Syn Meth, 2012, 3: 312–324
16 Liu H F, Hu Z, Haddadi H, et al. Hidden link prediction based on node centrality and weak ties. Europhys Lett, 2013, 101: 18004
17 Neville J, Jensen D. Relational dependency networks. J Mach Learn Res, 2007, 8: 653–692
18 Yu K, Chu W, Yu S P, et al. Stochastic relational models for discriminative link prediction. In: Proceedings of the 19th International Conference on Neural Information Processing Systems, 2007. 1553–1560
19 Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013. 3111–3119
20 Ou M D, Cui P, Pei J, et al. Asymmetric transitivity preserving graph embedding. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016. 1105–1114
21 Zhang Z W, Cui P, Zhu W W. Deep learning on graphs: a survey. IEEE Trans Knowl Data Eng, 2020. doi: 10.1109/TKDE.2020.2981333
22 Coscia M, Rossetti G, Giannotti F, et al. Uncovering hierarchical and overlapping communities with a local-first approach. ACM Trans Knowl Discov Data, 2014, 9: 1–27
23 Epasto A, Lattanzi S, Paes L R. Ego-splitting framework: from non-overlapping to overlapping clusters. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017. 145–154

24 Rees B S, Gallagher K B. Overlapping community detection by collective friendship group inference. In: Proceedings of International Conference on Advances in Social Networks Analysis and Mining (ASONAM), Denmark, 2010. 375–379
25 Jaccard P. The distribution of the flora in the alpine zone. New Phytol, 1912, 11: 37–50
26 Perozzi B, Al-Rfou R, Skiena S. Deepwalk: online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014. 701–710
27 Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013. 3111–3119
28 Grover A, Leskovec J. Node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International, 2016. 855–864
29 Tang J, Qu M, Wang M Z, et al. Line: large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web, 2015. 1067–1077
30 Wang H W, Wang J, Wang J L, et al. GraphGAN: graph representation learning with generative adversarial nets. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018. 2508–2515
31 Epasto A, Perozzi B. Is a single embedding enough? Learning node representations that capture multiple social contexts. In: Proceedings of the World Wide Web Conference, 2019. 394–404
32 Fawcett T. An introduction to ROC analysis. Pattern Recogn Lett, 2006, 27: 861–874
33 Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput, 1997, 9: 1735–1780
34 Zhou P, Shi W, Tian J, et al. Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016. 207–212
35 Cho K, van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014
36 Raghavan U N, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E, 2007, 76: 036106

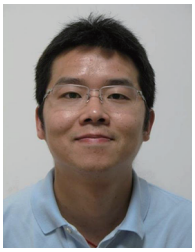
Network-splitter: a network feature extraction algorithm based on overlapping community and its application in link prediction

Hao LIAO[1], Xiaomin HUANG[1], Ziqiang WU[1], Mingyang ZHOU[1*], Rui MAO[1] & Binghong WANG[2]

1. *College of Computer Science and Software Engineering, Shenzhen University, Shenzhen* 518060*, China*;
2. *Department of Modern Physics, University of Science and Technology of China, Hefei* 230027*, China*
* Corresponding author. E-mail: zmy@szu.edu.cn

Abstract Link prediction is the task of forecasting the possibility of generating new links in a network based on network structure and node attributes. It is a basic problem in network science and valuable both in theory and practice. In recent years, deep learning methods are widely used for network representation to extract complex network features, which greatly improves the results of link prediction. The nodes in real-world networks have the phenomenon of local clustering; however, the current network representation learning methods focus on extracting the global features of the network, ignoring the local information features. In order to solve this problem, we propose a network-splitter model, which can learn the local feature representation of nodes in different communities. The model uses the idea of overlapping community to create a role copy of a node in each community and learns the feature representation of the role copy. Finally, the role copy information of the node in different communities is synthesized through the neural network. Using this method, both the global feature of the network and the local features of the nodes are summarized into the network representation, and then are applied to the link prediction. The experimental results show that the network-splitter model has strong competitiveness compared with the latest network representation learning methods.

Keywords link prediction, complex network, network representation learning, local node feature, overlapping community



Hao LIAO was born in 1987. He got his Ph.D. degree from Fribourg University, Switzerland in 2015. He is now a teacher at Shenzhen University. His main research interests include information networks, data mining, and complex networks.

Mingyang ZHOU was born in 1987. He got his bachelor and Ph.D. degrees in University of Science and Technology of China in 2010 and 2016 respectively. In the same year, he worked at Shenzhen University as an assistant professor. His research interests include complex networks, network control, and data mining.

Xiaomin HUANG was born in 1996. She got her master's degree from Shenzhen University in 2020. Her main research interests include information networks, data mining, and complex networks.

Ziqiang WU was born in 1997. He is now a graduate student at Shenzhen University. His main research interests include information networks, data mining, and complex networks.