一种大规模网络中基于节点结构特征映射的链接预测方法

李志宇 梁 循 周小平 张海燕 马跃峰
(中国人民大学信息学院计算机系 北京 100872)

摘 要 网络链接预测能够获取网络中丢失链接的重要信息或进行网络的动态演变分析.现有的基于节点相似性的网络链接预测方法往往针对简单的一(多)阶邻居信息或特定类型的小型网络,设计较为复杂的计算方法,其扩展性和大规模网络中的可计算性都受到了严峻的挑战.文中基于深度学习在神经网络语言模型中应用的启发,提出了一个 LsNet2Vec(Large-scale Network to Vector)模型.通过结合随机游走的网络数据集序列化方法,进行大规模的无监督机器学习,从而将网络中节点的结构特征信息映射到一个连续的、固定维度的实数向量.然后,使用学习到的节点结构特征向量,就可以迅速计算大规模网络中任意节点之间的相似度,以此来进行网络中的链接预测.通过在 16 个大规模真实数据集上和目前的多个基准的最优预测算法对比发现,LsNet2Vec 模型所得到的预测总体效果是最优的:在保证了大规模网络中链接预测计算可行性的同时,于多个数据集上相对已有方法呈现出较大的 *AUC* 值提升,最高达 8.9%.

关键词 链接预测;大规模网络;节点特征向量;连续性表达;神经网络;机器学习
中图法分类号 TP311 **DOI 号** 10.11897/SP.J.1016.2016.01947

A Link Prediction Method for Large-Scale Networks

LI Zhi-Yu LIANG Xun ZHOU Xiao-Ping ZHANG Hai-Yan MA Yue-Feng
(*School of Information, Renmin University of China, Beijing* 100872)

Abstract The problem of link prediction can be categorized into two classes, namely, missing links prediction and future links prediction. The former is the prediction of unknown links in sampling networks; and the other is the prediction of links that may exist in the future of evolving complex networks. Until now, most of the methods for link prediction are designed based on the assumption of node similarity, which defined by using the essential features of nodes. The similarity evaluation of two nodes making the sparsity and huge size of networks become two of the main challenges remain in link prediction problems. In this work, we present a new model, named LsNet2Vec, for link prediction in large-scale networks according to the unsupervised machine learning. The main idea of our method is embedding the features of nodes in large-scale networks into a lower and fixed dimension of vector in the set of real numbers. We conduct extensive experimental analysis on sixteen famous datasets and present a controlled comparison of the LsNet2Vec model against several strong baselines of link prediction methods,

收稿日期:2015-08-06;在线出版日期:2016-03-07.本课题得到国家自然科学基金(71271211,71531012)、北京市自然科学基金(4132067)、中国人民大学科学研究基金(10XNI029)、中国人民大学 2015 年度拔尖创新人才培育资助计划资助.**李志宇**,男,1991 年生,博士研究生,中国计算机学会(CCF)会员,主要研究方向为社会计算、机器学习.E-mail:zhiyulee@ruc.edu.cn.**梁 循**(通信作者),男,1965 年生,博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为神经网络、支持向量机、社会计算.E-mail:xliang@ruc.edu.cn.**周小平**,男,1985 年生,博士研究生,中国计算机学会(CCF)会员,主要研究方向为 Web 挖掘、社会计算.**张海燕**,女,1975 年生,博士研究生,中国计算机学会(CCF)会员,主要研究方向为复杂网络、社会计算、推荐系统.**马跃峰**,男,1976 年生,博士研究生,中国计算机学会(CCF)会员,主要研究方向为数据挖掘、机器学习与模式识别.

with *AUC* testing. Result show that our model performs comparably with state-of-the-art methods, such as Katz index and random walk restart method, in various experiment settings.

Keywords link prediction; large-scale networks; node feature vector; distributed representation; neural network; machine learning

1 引 言

网络链接预测(link prediction)是指利用已知的网络信息对未知的链接(existent yet unknown links)或者未来时间的链接(future links)进行预测[1]. 如图 1(A)所示,图 G 为已知的节点及其链接关系(实线),而 $\bar{G}$ 为 G 中不存在的链接关系(细虚线),则链接预测问题就是利用图 G 中的已知信息给出图 $\bar{G}$ 中细虚线的形成概率. 链接预测的相关研究在包括生物学领域、电子领域、信息领域等各类领域得到了广泛关注[2]. 典型的链接预测应用包括分子生物学的蛋白质交互关系预测[3-4]、合作关系预测[5-6]、社会网络关系预测[7-8]以及推荐系统的推荐预测[9]等等.

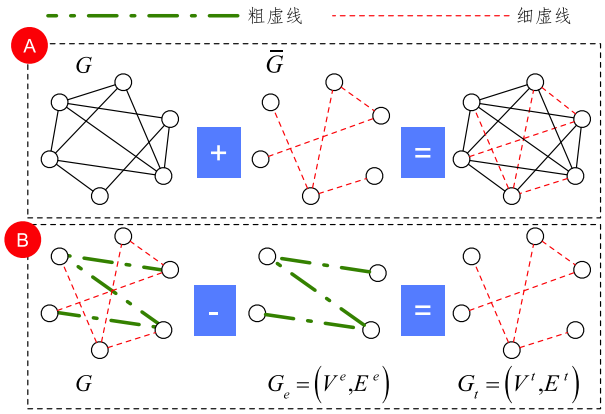


图 1 连接预测问题网络结构举例

对于网络链接的预测问题,从预测方法上来说,可以分为基于马尔可夫链的分析方法和基于机器学习的分析方法,其本质都是计算网络中两个节点存在链接的概率问题,即在表达形式上,可以根据特定的网络基本数据信息将链接关系转换为有向图链接和无向图链接,图中的节点用以表示网络中的节点,图中的边则表示链关系存在的可能性. 同时,对于有些特殊问题,图中的边还会赋予一定的权重用以表示不同节点边的重要性差异. 在存储数据上,通常将网络结构信息以邻接矩阵的形式存储,矩阵中的值既有可能是离散的(0,1)结构,也有可能是连续的概率分布. 然而,最终的链接预测问题都可以规范为:从网络节点的邻接矩阵 $\boldsymbol{A}$ 转换到邻接矩阵 $\boldsymbol{B}$ 的可能性问题.

从链接预测采用的信息源来划分,链接预测方法又主要可以分为基于网络拓扑结构的链接预测方法、基于网络内容属性的链接预测方法以及基于结构-属性的混合链接预测方法. 基于网络拓扑结构[3,10]的预测算法主要利用网络中节点的结构信息,包括节点的 N 阶邻居关系、节点的出入度信息等等,对节点之间的未知链接进行预测. 基于网络内容属性的链接预测[5]主要利用节点所包含的属性内容和标签信息,结合机器学习以及自然语言处理等工具,来对不同节点的相似性进行度量,从而预测节点之间构建链接的可能性. 最后,结构-属性的混合链接预测则是上述二者的综合与权衡.

一般而言,相对于现有的基于拓扑结构的预测算法,基于内容属性的相关算法虽然能够有效地利用节点的外部信息,使得在一定程度上提升算法的效果,但通常由于节点的属性信息获取较为困难,以及节点属性的信息的真实性无法得到保证,因此这类方法的稳定性需要进一步的衡量[11]. 近年来,基于拓扑结构的链接预测方法受到了越来越多的关注[3,10]. 相比节点属性方法而言,节点的结构信息可获取性较高,同时信息的真实性较强,因此能够具有更好的通用性.

目前,基于网络拓扑结构的主流预测算法包括基于节点 N 阶邻居信息的预测算法、基于最大似然估计的预测算法以及基于概率模型的预测算法. 在现有网络链接预测的研究中,有一个重要的应用思想和假设是:如果两个节点的相似性(结构相似性或者属性相似性)越高,则它们存在链接或者在未来形成链接的可能性则更高. 由此,在相似性假设的基础上产生了很多较为实用的链接预测方法. 其中,代表性的算法可分为 3 类[12]:基于网络局部信息的相似性算法[1]、基于网络节点路径相似性算法[13-14]以及基于网络随机游走的相似性算法[15-16]. 对比之下,虽然基于局部信息相似性的计算方法通常计算复杂度较低,但是计算的准确度却不如基于网络路径的相似性算法,而基于网络随机游走的相似性则是上述二者的折中,同时考虑计算效率以及预测的准确率.

随着大数据时代的来临，各类网络规模都在迅速的增加，并且呈动态变化，动辄上百万甚至上千万的网络数据使得现有的网络分析与链接预测方法变得捉襟见肘.因此，在原始小型网络环境中设计的相关链接预测算法在计算的可行性以及准确度上需要被重新设计.换而言之，需要针对现有大规模的网络结构特点，设计新的链接预测算法：在首先保证计算可行性的前提下，提高网络链接预测的准确率.

近年来，在机器学习领域，深度学习是一个值得重点关注的研究方法.深度学习已经在包括计算机视觉、音频处理以及自然语言处理等领域取得了巨大的成功[17].其中，在自然语言处理领域中，基于神经网络的语义空间模型的文本分布式表达的相关模型及其拓展得到了广泛的研究[18-22]，并且取得了较大的进展.词语特征的分布式表达模型的核心思想是将词语的语法或者语义特征映射到一个固定维度的连续空间，以此解决原有方法中存在的词语矩阵所包含的稀疏性问题以及计算的维数灾难[23].后续研究中，Mikolov 等人[24]在此基础上提出了 CBOW 模型以及 Skip-gram 模型用以大规模的文本连续性表达的学习，并在多个应用上取得了较好的结果.

本文的模型设计受到深度学习在自然语言中相关应用的启发，结合了网络随机游走的序列化方法，通过将传统网络中节点的结构特征映射到连续的、固定维度的向量空间，由此得到大规模网络中节点结构特征的分布式表达(节点向量)，从而应用于网络链接预测问题中.在链接预测问题中，本文采用的假设同样是基于节点相似性预测，即通过计算不同节点结构特征向量的余弦相似性估计节点之间存在链接的可能性.

和以往研究不同的是，对于大规模网络中节点的结构特征的获取不再是进行人工的构建，而是利用节点在网络中随机游走的方法，使用 LsNet2Vec 模型进行自动的无监督学习，从而有效避免了繁杂的手工网络特征构造工作，同时提高了算法在不同类型网络之间迁移的能力.实验证明，LsNet2Vec 模型拥有强大的数据扩展能力，能快速实现百万(笔记本电脑/数小时训练时间)甚至千万级别(笔记本电脑/数天训练时间)网络中节点结构特征的分布式表达的学习问题.

本文的主要贡献包括 3 个方面：

(1)在准确性方面：通过在广泛和大规模的真实数据集上的实证实验表明，LsNet2Vec 模型较大地提升了现有基于网络拓扑结构节点相似性用以大规模网络中链接预测问题方法的准确性.

(2)在计算效率方面：LsNet2Vec 模型训练得到的节点的分布式表达(节点向量)克服了在大规模网络中利用网络拓扑结构进行链接预测所带来的高计算复杂度问题，使得在大规模网络中对任意节点之间的相似性计算变为可行同时高效.

(3)在应用拓展方面：LsNet2Vec 模型所训练得到的节点结构特征向量为复杂网络中以网络结构特征基础的其他应用问题如社区发现、关键路径预测、跨平台用户检测等提供了一个可能的新思路和新方法.

本文的第 2 节主要介绍链接预测问题的相关工作，同时指出当前链接预测问题存在的难点；第 3 节对模型论述过程中涉及到的相关概念和符号进行详细的定义，同时给出链接预测以及模型评估问题的数学形式化表述；第 4 节对模型的主要框架以及模型优化涉及到的数学过程进行推导，同时给出参数修改的数学表达式以及最终算法的伪代码叙述；第 5 节给出实验的基本设置以及数据集、基准对比方法和评价指标的选取，对实验取得的相关结果进行分析并讨论超参对于实验结果的影响；最后，我们在第 6 节给出相应的研究结论以及下一步的研究展望.

2　相关工作

较为系统的链接预测问题最早由 Liben-Nowell 等人[25]提出，此后便得到了学术界的广泛关注.在文献[25]中，作者基于节点的相似性假设，对科学合作网络的链接预测问题进行了实证研究，同时作者提出了一些基本的节点相似性计算方法，这些方法也成为后续研究的对比基准.Liben-Nowell 等人提出的基准方法包括节点共同邻居相似性(Common Neighbor，CN)(局部信息)、Adamic-Adar(AA)相似性(局部信息)、Jaccard 系数相似性(局部信息)、Katz 相似性(全局路径信息)、SimRank 相似性(随机游走)等.此后，国内外学者围绕基于节点相似性的链接预测问题提出了各种类型的改进模型和链接方法，不断的从预测的准确性上进行相应的改善，具有代表性研究包括：

(1)基于局部信息.Zhou 等人[1]基于网络中资源分配(Resource Allocation，RA)的角度提出了一种新的节点相似性度量方法.其核心思想上假设网

络中每个单元都有一个资源,对于原始网络中某两个不存在链接的节点 A 和 B,其共同邻居为 A/B 间资源传递的媒介,假设每个媒介都会均匀的把资源传给它的邻居,那么 B 接收到的资源数就被定义为 A/B 节点之间的相似度. RA 方法虽然能够有效惩罚大度节点,但是当网络节点的平均度较小时,RA 算法和 AA 算法差别并不大. Liu 等人[26]基于社会网络定义不同社区成员之间的距离关系问题,结合多维度的社会距离表示,利用 Hash 方法在多维度的空间中进行节点相似度的计算,通过统计节点的局部结构特征,来对全局环境中的信息传递进行动态预测.

(2)基于路径信息. Lü 等人[27]提出了一种基于两个节点之间的局部路径(Local Path,LP)的相似性度量方法,该方法是在邻居节点的基础上进一步考虑了三阶邻居的贡献,从而挖掘节点之间的相似性. LP 方法是 CN 方法的扩展,当进一步考虑 N 阶扩展时,该方法则相当于 Katz 方法,复杂性较高. 同时,Lichtenwalter 等人[28]提出了一种有监督的加权的节点相似性度量方法,该方法在构建训练集时需要大量可靠标签,因此对于不同的未标注网络扩展性较差.

(3)基于随机游走. 这类方法是后续改进方法对效果和效率同时进行权衡的一类方法,其中平均通勤时间(Average Commute Time,ACT)、随机游走余弦相似性(Cos+)、有重启的随机游走(Random Walk with Restart,RWR)成为了基准的基于随机游走的相似性度量对照方法. 此后 Liu 等人[15]提出了一种基于叠加的局部随机游走相似性度量方法(Superposed Random Walk,SRW),它是在局部随机游走的基础上对 T 步及其前序结果进行加和从而得到 SRW 相似度. 该方法是对 RWR 的方法的改进,但只考虑了真实网络中的局域性特点(目的是为了减少计算复杂度),忽视了网络的全局特征,精度有所下降.

除上述网络节点相似度计算方法外,国内的其他相关研究包括:黄立威等人[29]针对异质信息网络的特征,使用元路径描述节点之间不同类型的关系,从而提出了关于异质信息网络的链路预测模型,通过组合不同元路径上对象之间的连接建立的概率来进行链路预测,取得了不错的效果. 刘冶等人[30]提出了一种基于低秩和稀疏矩阵分解的多源融合去噪链接预测算法,通过将主数据源和附加数据源进行有效融合,然后作为传统无监督拓扑链接预测算法的输入,来改进传统无监督预测算法的准确率. 吴祖峰等人[31]基于 AdaBoost 算法对基于网络拓扑结构的链接预测问题中存在的召回率低的问题进行了改进. 按照网络中节点之间是否存在链接关系,将链路预测问题定义为二分类问题,然后进一步遵循算法互补的原则选择若干具有代表性的链路预测算法作为弱分类器实现了链接预测. 李玉华等人[32]针对基于拓扑网络结构的链路预测方法中不存在时间属性的问题,结合科研合作网的特点,提出了一种基于链接重要性的动态链接预测方法. 通过引入链接重要性的度量,对拓扑属性和语义相似度等属性进行修正,以此考虑动态性以反映时间因素对链接形成的影响,最后利用分类技术进行预测.

经过已有文献的实验表明[11,33-35],现有方法中基于全局的节点信息的相似度计算方法,如 Katz 方法、LHN-II 方法,以及基于随机游走方法中,如 ACT 方法、Cos+方法、RWR 方法、SimRank 方法等在大规模网络中的计算复杂度都很高,同时计算效果的表现也不是很稳定. 而目前,专门针对大规模网络(10^5 个节点以上)的链接预测算法几乎很少,目前采用的做法是针对现有的链接预测算法设计并行的思路进行改进,例如 Ogata 等人[36]提出了一种基于乔里斯基分解(Cholesky decomposition)的矩阵分解方法,来对链接的传播机制进行分析. 然而,虽然这种方法能够相对于传统的方法在速度上有所提升,但是在准确度上却不如以前. 还有一种做法就是基于分层或者社区标注的思想,进行预处理后在通过社区标签或者分层标签进行链接预测. 例如 Shin 等人[37]提出了一个基于树结构的,通过结合层次聚类的方式在多个数量级上进行链接预测. 然而,由于作者在模型中使用的是一个平衡层次树结构来对现有的数据进行拟合,使得相对来说并不能够有效地切合很多实际数据集中所存在的情况.

但是上述策略始终是无法针对性地解决大规模网络所带来的数据稀疏性以及邻接矩阵的维数灾难问题. 因此,本文针对现有研究存在的计算复杂度高,需要的内存开销大等问题提出了基于节点结构特征的分布式表达模型 LsNet2Vec.

3 符号及问题定义

本节首先给出了模型中涉及到的基础符号定义(如表 1),然后定义了模型中涉及到的重要概念,最后给出了链接预测问题求解思路的形式化表达.

表 1　基础符号定义

符号	定义
$G=(V,E)$	图 G 包含节点集合 V，以及边集合 E
$\|G\|$	图 G 中包含的节点数目
$\bar{G}$	图 G 的补图
$X\in V$	正体 X 表示图 G 节点集合 V 中的节点 X
m	特征向量的维度数，正整数值，通常的 $0<m\leq 1000$
$\boldsymbol{M}$	节点特征嵌入矩阵 $\boldsymbol{M}$，是一个 $\|G\|\times m$ 实数值矩阵
$S_{x,y}^{\text{Method}}$	相识度计算方法 Method 基于节点 x,y 的相似度得分

定义 1.　训练预测窗口 *Window N*.

在 LsNet2Vec 模型中，预测窗口 *Window N* ($N\geq 4$)为一个实值正整数. 其意义为限制每一个被预测节点 S 的周边节点 $Around(S)$ 的覆盖幅度，数值上等于 $Around(S)$ 中节点的个数.

定义 2.　节点 S 的周边节点 $Around(S)$.

和现有网络分析中关于邻居节点的定义不同的是，LsNet2Vec 模型中的邻居节点是根据网络节点序列化(4.2 节)的结果动态变化的. 因此，对于同一个节点，在不同的输出序列化训练集中，甚至不同的训练窗口中，其周边节点 $Around(S)$ 是不同的，并且 $Around(S)$ 可以包含节点 S 最多任意预测窗口 *Window N* 大小阶数的邻居. $Around(S)$ 的形成方式为：

对于节点 S，首先随机向前选取 *step* 个节点，然后随机向后选取 *N-step* 个节点构成，其中每次向前选取的随机值 *step* 满足：$0\leq step\leq N$.

以图 2 中节点 G 为例，节点 G 出现在了前 3 条训练集中，那么此时以 G 为预测目标的，即 *Window* $N=4$ 时的可能周边节点为

G 在第 1 条训练集中的周边节点($step=3$)：

$$Around(G)=\{F,D,B,H\}\rightarrow G;$$

G 在第 2 条训练集中的周边节点($step=2$)：

$$Around(G)=\{C,H,B,F\}\rightarrow G;$$

G 在第 3 条训练集中的周边节点($step=0$)：

$$Around(G)=\{B,C,E,H\}\rightarrow G.$$

其中，节点 D,B,H,E 为节点 G 的直接邻居(一阶邻居)，节点 C,F 为节点 G 的间接邻居(二阶邻居).

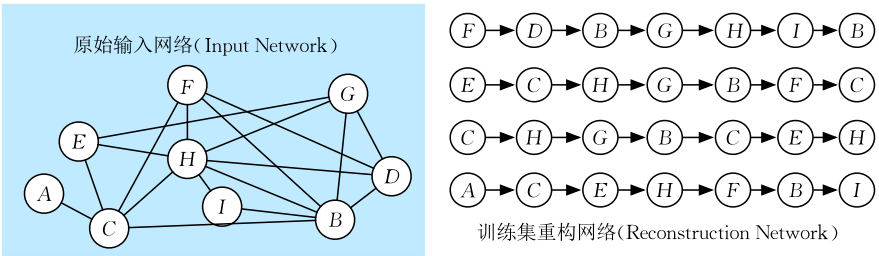


图 2　*MAX_LENGTH*=7，*WALK_TIMES*=4 的随机游走网络节点序列化结果

定义 3.　节点特征分布式表达嵌入矩阵 $\boldsymbol{M}$.

实质上，嵌入矩阵 $\boldsymbol{M}$ 可以看作一个关于网络中节点特征向量的“表”结构，每个节点对应嵌入矩阵 $\boldsymbol{M}$ 的一行，每行的维度数即节点特征向量的维度数 m，每次获取特定节点特征向量的过程即为一次查表过程.

问题 1. 网络链接预测.

给定一个无向图 $G=(V,E)$，得到 $\bar{G}$，如图 1(A)所示. 则对于既定的链路预测方法，得出补图 $\bar{G}$ 中每两个节点(每条边)之间的分值(或概率) $Sim(x,y)$，然后对所有分值进行排序，得到最可能的边分布情况. 即链接预测问题的目标是给出补图 $\bar{G}$ 中边的形成概率.

问题 2. 网络链接预测的评估方法.

为了评估链接预测算法的准确性，通常，对于已知图 $G=(V,E)$ 中的链接集合 E，将其拆分为训练集 E^t 和测试集合 E^e，其中，拆分需要满足：$E=E^t\cup E^e$，同时 $E^t\cap E^e=\varnothing$，如图 1(B)所示. 即将测试集 E^t 中的信息看作已知信息，以此对测试集合 E^e 中存在的链接进行预测.

4　LsNet2Vec 模型

4.1　模型架构概述

如图 3 所示，LsNet2Vec 模型为 3 层架构，分别包括输入层、投影层以及输出层，本节将分别对每层架构的基本功能进行阐述.

4.1.1　输入层

输入层的主要功能是对原始网络进行序列化(即图 3 中 K1)处理. 序列化指的是按照一定的规则对网络中的节点进行遍历，然后按照序列化的格式进行输出，进而重构训练集. 目的是利用目标节点的周围环境特征来对目标节点进行预测，即通过节点环境结构特征的学习来预测节点自身. 训练集的重构是网络节点结构特征向量化的关键步骤之一，也是涉及节点结构特征提取好坏的重要预处理步骤.

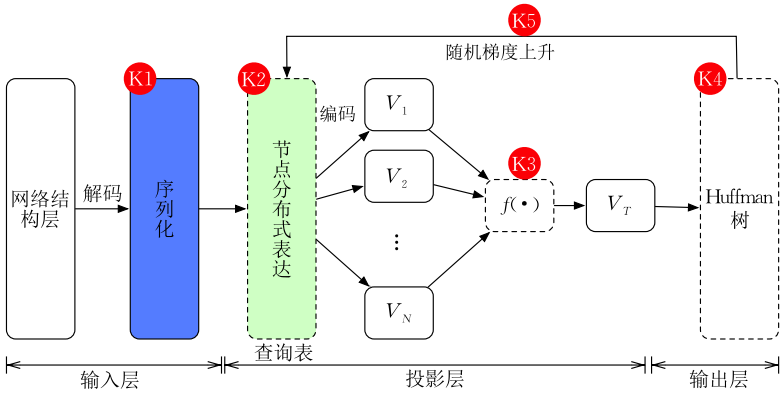


图 3 LsNet2Vec 模型架构

即对于网络中的某个节点 S,模型的基本目标是在已知节点 S 的周边节点 $Around(S)$ 的前提下,对节点 S 进行预测,如式(1)所示:

$$Training\ set=\{(S,Around(S))\},\ S\in V \quad (1)$$

则对所有预测目标的预测概率对数似然后,最终目标函数为

$$O=\arg\max\sum_{S\in V}\log p(S,Around(S)) \quad (2)$$

式(2)中,V 为重构后的训练集,$p(\cdot)$为概率函数.

4.1.2 投影层

投影层的主要功能是将序列化的训练集按照给定训练窗口大小 α 在嵌入矩阵 $\boldsymbol{M}$(即图 3 中 K2)中进行查找,然后将查找得到的向量集合作为参数传递给聚合函数 $f(\cdot)$进行处理,如图 4 所示. 向量维度数 m 为超参,通常由具体应用结合原始网络的大小以及训练精度和时间等要求动态设计,该超参对结节点特征提取用于链接预测问题的好坏影响将在第 5.5.1 节进行实验讨论.

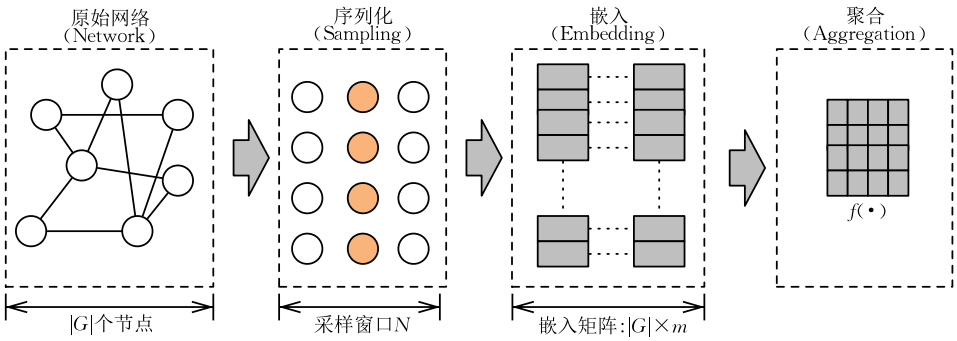


图 4 投影层结构关系

4.1.3 输出层

如图 5 所示,输出层的构成是一棵 Huffman 树,选用 Huffman 树对网络节点进行重构存储能够有效地降低计算的复杂度. LsNet2Vec 模型中 Huffman 树的编码原则是基于输入图中节点的度的大小进行编码存储. Huffman 树中共包含 $|G|$ 个叶子节点. 树中的每个节点可以看作一个分类器,对输入的信息进行分类预测,其详细功能在 4.3 节进行详细分析.

4.2 网络节点的序列化

网络节点的序列化是对原始网络节点的结构特征进行初步提取的基本步骤. 在 LsNet2Vec 模型中,我们采用随机游走的方法获取序列化的节点信息. 该步骤主要包含两个参数:每次游走的最大长度 MAX_LENGTH 和重构数据集的随机游走次数 $WALK\_TIMES$. 通过实验发现,上述参数和输入网络的节点数量和边数存在相应的数量关系. 一般来说,对于较大规模的网络,如果式(3)取值过小,则会导致网络结构初步提取不充分,从而使得训练得到的节点向量无法充分学习得到网络的结构特征.

$$MAX\_LENGTH\times WALK\_TIMES \quad (3)$$

反之,如果式(3)取值过大则会增加模型的训练时间和复杂度,甚至降低预测效率.

以图 2 为例,假设 $MAX\_LENGTH=7$,$WALK\_TIMES=4$,对原始输出网络进行一轮序列化. 其基本步骤为:每次游走时随机从网络中选取一个初始出发节点,然后在网络中进行随机游走,游走的步长为 $MAX\_LENGTH-1$,将走过的节点按顺序输出,作为一个训练序列,重复该步骤 $WALK\_TIMES$ 次. 对于参数 MAX_LENGTH 的分析在 5.5.2 节

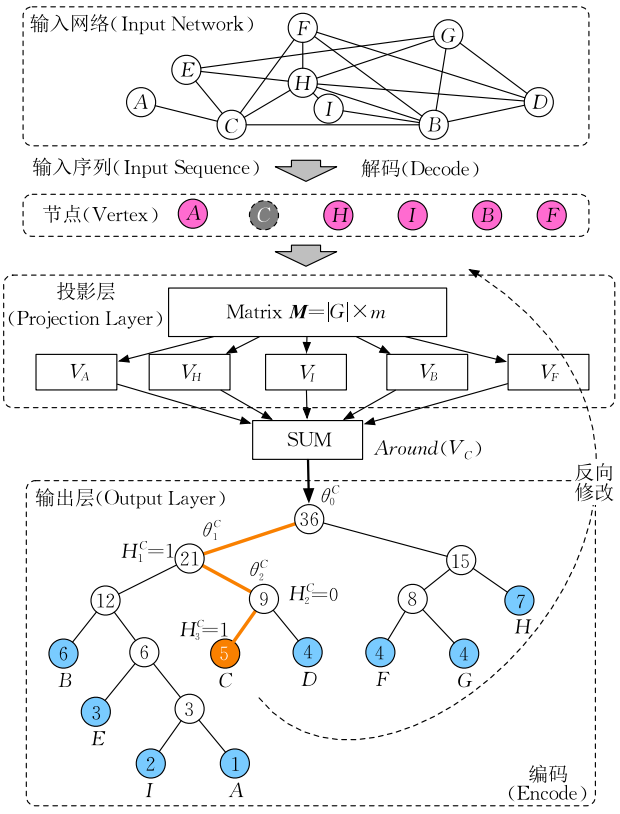


图 5　以叶子节点 C 为目标的一次训练过程

进行了详细的讨论.

4.3　模型推导与参数训练

本节中,将结合一个具体的例子,对模型参数的训练过程进行详细推导. 首先,给出了模型训练中的重要符号及其定义;然后,以一次训练过程为例,给出了分层的 softmax 方法;接着,结合目标函数,采用随机梯度上升的方法对参数进行寻优,并给出了节点向量,参数的具体更新过程.

4.3.1　目标函数推导

如图 5 中输出层所示. 考虑 Huffman 树中某个叶子节点对应图 G 节点集合中的某个节点 C,定义符号如下所示:

(1) $p^C=\{p_0^C,p_1^C,\cdots,p_n^C\}$:从根节点出发到达叶子节点 C 的所包含的路径,其中 p_0^C 为 Huffman 树的根节点,p_n^C 为对应的叶子节点 C,$|p^C|$ 为路径中包含的节点个数;

(2) $\{H_1^C,H_2^C,\cdots,H_n^C\}$:叶子节点 C 对应的 Huffman 编码;

(3) $\{\theta_0^C,\theta_1^C,\cdots,\theta_{n-1}^C\}$:到达叶子节点 C 路径中所包含非叶子节点的参数向量集合;

在 LsNet2Vec 模型中,对于节点 C 的预测同样基于 Markov 假设,即对于一个节点 C,其出现的概率只与其周边的 n 个邻居相关,这些邻居既有可能是节点 C 的直接邻居,也有可能是其间接邻居,如社会网络中,朋友的朋友关系.

现考虑,以输入序列 $\{A,H,I,B,F\}\rightarrow C$ 进行的一次训练和参数调整的过程:

对于输入序列 $\{A,H,I,B,F\}$,一次模型的训练目标是通过 Huffman 树的分类,以最大的概率到达 C 所在的叶子节点. 对于叶子节点 C,其 Huffman 编码为式(4):

$$\{H_1^C,H_2^C,H_3^C\}=\{1,0,1\} \tag{4}$$

此时该编码路径上所包含的非叶子节点的参数向量集合为式(5):

$$\{\theta_0^C,\theta_1^C,\theta_2^C\} \tag{5}$$

由图 5 中投影层可知,对于输入序列 $\{A,H,I,B,F\}$ 在节点嵌入矩阵 $\boldsymbol{M}$ 中进行查找,可以得到序列节点的分布式表达:

$$\{A,H,I,B,F\}\rightarrow\{v(A),v(E),v(I),v(B),v(F)\} \tag{6}$$

然后对式(6)通过聚合函数 $f(\cdot)$ 进行聚合,得到结构特征环境向量 $\boldsymbol{v}_{A(C)}$,本模型中我们选用基本的叠加函数进行计算,如式(7)所示.

$$\boldsymbol{v}_{A(C)}=Around(v_C)=\sum_{i\in Around(C)}\boldsymbol{v}_i \tag{7}$$

其中,$Around(C)$ 为训练窗口,在本例子中窗口大小为 5,即选取以预测目标 C 为中心的前后,窗口大小为 5 的邻居节点作为 C 的周边结构特点的描述. 此时预测问题转换为:以向量 $\boldsymbol{v}_{A(C)}$ 根节点的输入,如何计算到达叶子节点 C 的概率.

如 4.1.3 节所述,将 Huffman 树中的每个节点看作一个分类器,我们选取经典的 sigmoid 函数作为的激活函数结合逻辑回归,那么对于输入向量 $\boldsymbol{X}$,其输出为正例的概率为式(8):

$$\sigma(\boldsymbol{X}^{\mathrm{T}}\theta)=\frac{1}{1+e^{-\boldsymbol{X}^{\mathrm{T}}\theta}} \tag{8}$$

输出为负例的概率为 $1-\sigma(\boldsymbol{X}^{\mathrm{T}}\theta)$.

那么,对于叶子节点 C 而言,其 Huffman 编码 $\{1,0,1\}$ 对应的分类过程为式(9):

$$\begin{cases}1\rightarrow p(1\,|\,\boldsymbol{v}_{A(C)}^{\mathrm{T}},\theta_0^C)=\sigma(\boldsymbol{v}_{A(C)}^{\mathrm{T}}\theta_0^C)\\0\rightarrow p(0\,|\,\boldsymbol{v}_{A(C)}^{\mathrm{T}},\theta_1^C)=1-\sigma(\boldsymbol{v}_{A(C)}^{\mathrm{T}}\theta_1^C)\\1\rightarrow p(1\,|\,\boldsymbol{v}_{A(C)}^{\mathrm{T}},\theta_2^C)=\sigma(\boldsymbol{v}_{A(C)}^{\mathrm{T}}\theta_2^C)\end{cases} \tag{9}$$

因此,基于邻居节点的向量的输入成功到达叶子节点 C 的最终联合概率为式(10):

$$p(C\,|\,Around(C))=\prod_{i=1}^{3}p(H_i^C\,|\,\boldsymbol{v}_{(C)}^{\mathrm{T}},\theta_{i-1}^C) \tag{10}$$

将上述推导过程一般化,得到任意某个叶子节点 S,其输入邻居聚合向量为 $\boldsymbol{v}_{A(S)}$,则从根节点到叶子节点,正确分类的条件概率为式(11):

$$p(S|Around(S))=\prod_{j=1}^{|p^S|}p(H_j^S|\boldsymbol{v}_{A(S)}^{\mathrm{T}},\theta_{j-1}^S) \tag{11}$$

其中：

$$p(H_j^S|\boldsymbol{v}_{A(S)}^{\mathrm{T}},\theta_{j-1}^S)=\begin{cases}\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S), & H_j^S=1\\ 1-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S), & H_j^S=0\end{cases} \tag{12}$$

联合式(12)、(11)代入目标似然函数式(2)中得到

$$\begin{aligned}O&=\arg\max\sum_{S\in V}\log\prod_{j=1}^{|p^S|}p(H_j^S|\boldsymbol{v}_{A(S)}^{\mathrm{T}},\theta_{j-1}^S)\\&=\arg\max\sum_{S\in V}\log\prod_{j=1}^{|p^S|}\{[\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]^{H_j^S}\cdot\\&\quad[1-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]^{(1-H_j^S)}\}\\&=\arg\max\sum_{S\in V}\sum_{j=1}^{|p^S|}\{H_j^S\cdot\log[\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]+\\&\quad(1-H_j^S)\cdot\log[1-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]\}\end{aligned} \tag{13}$$

即式(13)为需要优化的目标函数.

4.3.2 参数估计

由式(13)可知，记

$$\begin{aligned}O(S,j)=&H_j^S\cdot\log[\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]+\\&(1-H_j^S)\cdot\log[1-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]\end{aligned} \tag{14}$$

与文献[24]所提出的连续词袋模型相似，针对上述目标函数同样采用随机梯度上升进行优化，即每次选取一个序列化后的窗口样本$(S,Around(S))$，就对目标函数的参数进行一次更新. 那么，针对目标函数式(14)中存在的参数：$\boldsymbol{v}_{A(S)}$和θ_{j-1}^S做出更新即可.

因此分别对上述参数关于目标函数求偏导数，得出参数的更新梯度：

(1) 参数θ_{j-1}^S更新如式(15)所示：

$$\begin{aligned}\frac{\partial O(S,j)}{\partial\theta_{j-1}^S}&=\{H_j^S\cdot[1-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]-\\&\quad(1-H_j^S)[\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]\}\cdot\boldsymbol{v}_{A(S)}^{\mathrm{T}}\\&=\{H_j^S-H_j^S\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)+\\&\quad H_j^S\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)\}\cdot\boldsymbol{v}_{A(S)}^{\mathrm{T}}\\&=[H_j^S-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]\cdot\boldsymbol{v}_{A(S)}^{\mathrm{T}}\end{aligned} \tag{15}$$

因此，参数θ_{j-1}^S的更新梯度如式(16)所示：

$$\theta_{j-1}^S:=\theta_{j-1}^S+\eta[H_j^S-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]\cdot\boldsymbol{v}_{A(S)}^{\mathrm{T}} \tag{16}$$

式(16)中，η用以控制学习速率.

(1) 参数$\boldsymbol{v}_{A(S)}$的更新如式(17)所示：

$$\begin{aligned}\frac{\partial O(S,j)}{\partial\boldsymbol{v}_{A(S)}}&=\{H_j^S\cdot[1-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]-\\&\quad(1-H_j^S)[\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]\}\cdot\theta_{j-1}^S\\&=\{H_j^S-H_j^S\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)+\\&\quad H_j^S\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)\}\cdot\theta_{j-1}^S\\&=[H_j^S-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]\cdot\theta_{j-1}^S\end{aligned} \tag{17}$$

由于$\boldsymbol{v}_{A(S)}$为节点S的邻居节点通过加法聚合而成，因此关于$\boldsymbol{v}_{A(S)}$的梯度更新可以直接反馈到其邻居节点向量的分布式表达上去，因此，对于$i\in Around(S)$的每个节点向量$\boldsymbol{v}_i$，其更新梯度为式(18)：

$$\boldsymbol{v}_i:=\boldsymbol{v}_i+\eta\sum_{j=1}^{|p^S|}\frac{\partial O(S,j)}{\partial\boldsymbol{v}_{A(S)}} \tag{18}$$

即节点特征向量的更新梯度为式(19)：

$$\boldsymbol{v}_i:=\boldsymbol{v}_i+\eta\sum_{j=1}^{|p^S|}[H_j^S-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]\cdot\theta_{j-1}^S \tag{19}$$

4.3.3 模型小结

通过将模型在重构的网络节点关系集上进行遍历，对模型中的各个参数以及节点向量进行调整，算法1给出LsNet2Vec模型的基本流程.

算法1. LsNet2Vec模型的训练与节点向量构建.

输入：网络节点关系(边序列)

输出：每个节点的连续性向量表达$\boldsymbol{v}_i$

BEGIN

training sets reconstruction: Random walk with *Window size* N in edge list, generate training pairs: $(S,Around(S))$

parameters initialization: Parameters θ in Huffman tree and $\boldsymbol{v}_i$ in Matrix $\boldsymbol{M}$

FOREACH $(S,Aournd(S))$ in training sets DO

Initialization: $q=0$, $\boldsymbol{v}_{(S)}=\sum_{i\in Around(S)}\boldsymbol{v}_i$

FOREACH non-leaf node in Path DO

Update:

$q:=q+\eta[H_j^S-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]\cdot\theta_{j-1}^S$

Update:

$\theta_{j-1}^S=\theta_{j-1}^S+\eta[H_j^S-\sigma(\boldsymbol{v}_{A(S)}^{\mathrm{T}}\theta_{j-1}^S)]\cdot\boldsymbol{v}_{A(S)}^{\mathrm{T}}$

END

FOREACH node in $Around(S)$ DO

Update: $\boldsymbol{v}_i:=\boldsymbol{v}_i+q$

END

END

END

LsNet2Vec模型采用随机梯度上升对参数进行优化，模型收敛与训练集重构的方式相关联. 在LsNet2Vec模型中，我们通过随机初始化参数后，在重构的训练集中用随机无放回选取训练对$(S,Around(S))$的方式进行参数训练，直到训练集中所有训练对被随机遍历完毕后停止训练. 在我们的实际实验过程中，我们将LsNet2Vec模型的迭代隐含到训练集的重构中，即可以通过增大随机游走的次数或者随机游走的步长来构建更多的训练样例，以达到模型迭代的目的. 在实际实验过程中，我们对学习速率采用的是线性降低的方式，随着数据集遍历

的进行，学习速率逐步降低，最后训练结束时学习速率降为 0.

4.4　节点向量与链接预测

训练得到节点结构特征的分布式表达 $\boldsymbol{v}_i$ 之后，就可以将原有网络节点拓扑结构的相似性转换为向量相似度计算问题了. 为了降低计算复杂度，增强模型在大规模网络中的拓展和计算能力，我们选取被广泛采用的余弦相似度来计算不同节点之间的相似性，即

$$S_{x,y}^{\mathrm{N2V}}=\cos(v_x,v_y)=\frac{v_x\cdot v_y}{\|v_x\|\cdot\|v_y\|} \tag{20}$$

因此，通过训练一次得到的节点结构特征向量后，就可以重复查询使用，结合式(20)快速计算得到大规模网络里面任意两个节点之间的结构相似度.

4.5　算法复杂度分析与对比

4.5.1　时间复杂度分析

LsNet2Vec 模型训练节点特征向量的复杂度如式(21)所示：

$$T_{\mathrm{N2V}}=N\times m+m\times\log_2(|G|) \tag{21}$$

式(21)中，$|G|$ 为图中节点的数量，m 为节点特征向量的维度，N 为随机采样节点窗口的大小. 因此，在模型向量的训练阶段，LsNet2Vec 模型训练的复杂度为 $O(\log_2(|G|))$，相对于 Shin 等人[37]提出的 MSLP 算法的 $O(|G|)$时间开销要低.

同时，如果需要计算整个网络中所有节点之间的相似度，则 LsNet2Vec 模型的复杂度为 $T_{\mathrm{N2V}}+O(|G|^2)$，即 $O(|G|^2)$，因此从计算复杂度上来讲，模型的复杂度和最近邻类方法中最简单的 CN 算法是相同的，因此相对其他的基于路径或者全局的计算方法而言，如 RWR 的 $O(|G|^3)$的复杂度，LsNet2Vec 模型的复杂度同样是较低的.

4.5.2　空间复杂度分析

不同于以往的链接预测的存储或计算策略，LsNet2Vec 模型是基于内存友好的方式构建的. 首先，LsNet2Vec 模型只需要存储节点的特征向量矩阵，即空间开销为 $|G|\times m$；而对于传统的计算方法，在计算过程中通常依赖于邻接矩阵和相似度矩阵的计算方式，其内存开销至少是 $|G|\times|G|$. 对于大规模的网络，LsNet2Vec 模型的节点特征向量维度通常设置为 $m\in[100,1000]\ll|G|$，因此在训练时模型的空间开销远远低于以往的邻接矩阵式计算方法.

其次，LsNet2Vec 模型采用的是在线学习的策略，对于新增加的节点，通过在节点附近按照一定方法进行重新采样，可以在原有模型的基础上继续训练，从而能够有效地适应动态的，快速增长的网络节点特征学习的问题.

5　实验与分析

5.1　数据集

我们选取了公路网络、合作网络、购买共现网络、社会网络、邮件通信网络、维基百科网络以及蛋白质互作用网络 6 个大类，共 16 个数据集，其拓扑汇总信息如表 2 所示. 这些数据集都属于较大规模的网络数据集，如公路网络(276 万条边)、Youtube 社交网络(298 万条边)等. 数据主要来源于斯坦福大学网络分析项目组(Stanford Network Analysis Project，SNAP)①的公开网络项目，下面分别对这些数据集的构成意义以及拓扑特性进行简要介绍.

表 2　对比数据集基本拓扑信息

类型	数据集(缩写)	点数	边数	密度	平均度	平均集聚度
公路网络	Pennsylvania PRN	1.09E+06	1.54E+06	2.60E−06	2.834	0.047
	Texas TRN	1.38E+06	1.92E+06	2.02E−06	2.785	0.047
	California CRN	1.97E+06	2.77E+06	1.43E−06	2.815	0.046
合作网络	Astro ACN	1.88E+04	1.98E+05	1.12E−03	21.106	0.630
	High-Energy HCN	1.20E+04	1.19E+05	1.64E−03	19.740	0.611
	Condense CCN	2.31E+04	1.87E+05	3.49E−04	8.083	0.633
	DBLP DCN	3.17E+05	1.05E+06	2.08E−05	6.622	0.632
购买网络	Amazon APN1	3.35E+05	9.26E+05	1.65E−05	5.529	0.396
	Amazon APN2	4.01E+05	3.20E+06	2.92E−05	11.728	0.402
社交网络	Youtube YSN	1.13E+06	2.99E+06	4.64E−06	5.265	0.081
	Epinions ESN	7.59E+04	5.09E+05	1.41E−04	10.694	0.137
	Slashdo SSN	7.74E+04	9.05E+05	1.82E−04	14.128	0.055
通信网络	Enron EEN	3.67E+04	1.84E+05	2.73E−04	10.020	0.497
	EU UEN	2.65E+05	4.20E+05	1.04E−05	2.757	0.067
维基百科	Talk network WTN	2.39E+06	5.02E+06	1.63E−06	3.892	0.053
	Vote network WVN	7.12E+03	1.04E+05	3.98E−03	28.323	0.141

① http://snap.stanford.edu

5.1.1 公路网络

公路网络数据集主要包含 3 个子数据集:分别是 Pennsylvania、Texas 和 California,为无向图.其中,节点代表路的交叉点或者终点,同时这些终点或者交叉点又被不同的路所连接.在这 3 个数据集中,最大的数据集为 California,包含了 196 万个节点和 276 万条边,同时,3 个数据集的平均集聚系数[①]都是相似的,约为 0.0465.

5.1.2 作者合作网络

作者合作网络主要包含 3 个子数据集:Arxiv Astro、High Energy、Condense Matter Physics 合作网络与 DBLP 合作网络,为无向图.这些数据集都是以作者为节点,以作者之间的论文共现为边的形成条件,即如果两个作者共同出现在一篇论文中,则节点之间形成一条边,如果 k 个作者同时出现在一篇论文中,则形成一个完全图,即每个节点之间两两相连.通过统计发现,作者合作网络的平均集聚系数都较高,约为 0.627.

5.1.3 购买共现网络

购买共现网络包含两个数据集,都来自亚马逊网站上的商品信息,为无向图.它是根据亚马逊网站上关于"Customers Who Bought This Item Also Bought"的推荐信息来进行抓取的.类似的,以商品为节点,如果两个商品同时出现在一次推荐信息中,则节点之间形成一条边.购买共现网络的平均集聚系数相对合作者网络而言稍低,约为 0.395.

5.1.4 社会网络

社会网络数据集包含 3 个子数据集:Youtube、Slashdo 和 Epinions.Youtube 是国外的著名视频分享社交网站,用户之间可以互相形成朋友关系,或者构建自己的圈子,其平均集聚度系数为 0.0808,该数据包含了 298 万条边,为一个较大型的社会网络.Slashdo 是一个技术行业新闻社交网络,它允许用户之间建立朋友或者敌人的关系,以此形成不同的社区,其平均集聚系数为 0.0549.Epinions 是一个在线点评网站,它允许用户之间建立互相信任的关系,即"who-trust-whom",其平均集聚系数为 0.1279.

5.1.5 邮件通信网络

邮件通信网络包含了 Enron 和 EU 两个较为著名的数据集.Enron 数据集包含了 3 万多个节点之间的互相通信关系.网络中每个节点代表一个账号,不同账号之间如果存在超过一次的邮件联系,则形成一条边,网络的平均集聚系数为 0.4970.EU 数据集则是由 European research institution 提供的在 18 个月期间的邮件账号的来往信息,其平均集聚系数为 0.0671.

5.1.6 维基百科网络

维基百科网络包含了 Wiki Talk 和 Wiki Vote 两个数据集.Wiki Talk 包含了约 500 万条边,每条边表示该用户节点和其他的用户至少存在存在一次沟通和交流,其平均集聚系数为 0.0526.Wiki Vote 为关于维基百科中管理员选举投票的数据集,每条边表示用户之间存在投票的行为,其平均集聚系数为 0.1409.

5.2 度量指标

常用的链接预测准确度的评价指标包括 *AUC*(Area Under the Receiver Operating Characteristic Curve)和 *Precision* 两类.其中,*AUC* 主要侧重在整体上衡量算法的准确度,而 *Precision* 主要侧重在只评价部分链接(排序前 N 位)预测的准确率.和文献[29,38-39]等模型采用的评价方法一致,我们同样使用 *AUC* 作为模型的度量指标.

一般而言,*AUC* 评价指标可以理解为一个概率值,其值表示在测试集 E^e 中节点链接的得分比一个随机选择实际上不存在的链接得分高的概率,一次测试 $Test_i$ 的具体计算方法如下:

(1)在测试集 E^e 中随机选出一个链接 E_x^e,同时在已知图 $G=(V,E)$ 的补图 $\bar{G}$ 中随机选择一个不存在的链接 $\bar{E}_y$.

(2)利用训练集得到的数据分别计算两条链接的形成概率:$p(E_x^e)$,$p(\bar{E}_y)$.

(3)如果 $p(E_x^e)>p(\bar{E}_y)$,$Test_i=1$;或者 $p(E_x^e)=p(\bar{E}_y)$,$Test_i=0.5$,否则 $Test_i=0$

因此,经过 n 次上述实验后,最终 *AUC* 的计算方法为式(22):

$$AUC=\frac{\sum_{i=1}^{n} Test_i}{n} \tag{22}$$

使用 *AUC* 评价指标来衡量链接预测准确率时,*AUC* 越接近于 1,则算法预测效果越好.通常而言,如果某个算法的 *AUC* 值小于 0.5,则说明该算法的效果非常差,甚至比随机的链接预测方法都差.

5.3 基准方法

基准对比方法主要涉及三大类,分别是局部基准方法(CN/RA/AA),路径基准方法(Katz/LP)以及随机游走基准方法(ACT/RWR).上述方法的选

① https://en.wikipedia.org/wiki/Clustering_coefficient

取主要考虑到两个方面：(1)权威性. 上述方法无论是在大规模的链接预测研究，还是小规模的链接研究中都被公认为较好的参照方法；(2)有效性. 上述方法在已有文献的众多对比实验中都能够在特定的数据集上取得最佳 *AUC* 值，因此适合作为模型可扩展性检验的标准参照方法. 下面，将对上述方法作简单阐述.

5.3.1　基于共同邻居的相似度计算(CN)

在网络分析算法中，节点共同邻居的相似度计算方法是最简单的基于局部信息的相似性定义方法[40]. 该算法基于的假设是两个节点的公共节点数越多，则它们之间的稳定性越强，由此相似性也就越高. 因此在链接预测的问题中，两个节点共同邻居节点数越多，则两者之间存在相互链接的可能性越大. CN 算法中，节点的相似性计算方法如式(23)所示：

$$S_{x,y}^{\mathrm{CN}}=|N(x)\cap N(y)| \tag{23}$$

5.3.2　基于资源分配算法的相似度计算(RA)

如相关工作一节所述，资源分配的相似度计算方法主要是以资源的重新分配为基本思路. RA 算法对于每个媒介的传递资源以 $1/K$ 的形式递减. 其相似度的计算方法如式(24)所示：

$$S_{x,y}^{\mathrm{RA}}=\sum_{Z\in N(x)\cap N(y)}\frac{1}{k(Z)} \tag{24}$$

5.3.3　基于 Adamic-Adar 算法的相似度计算(AA)

AA 算法[41]同时考虑了两个共同节点度的信息，核心思想是提高度小的共同邻居节点的贡献值. 例如在商品共现问题中，两个共同用户同时购买了冷门商品的相似性往往比同时购买热门商品的用户更相似. AA 算法通过共同邻居节点的度为每个节点赋予一个权重值，该权重值等于节点度的对数分之一，其计算定义为式(25)：

$$S_{x,y}^{\mathrm{AA}}=\sum_{Z\in N(x)\cap N(y)}\frac{1}{\log k(Z)} \tag{25}$$

5.3.4　基于 Katz 方法的相似度计算(KA)

Katz 方法考虑节点 (x,y) 之间所有的路径数，且对于较短的路径赋予叫大的权重，而较长的路径赋予较小的权重，它的计算方法如式(26)所示：

$$S_{x,y}^{\mathrm{Katz}}=\beta\boldsymbol{A}+\beta^2A^2+\beta^3A^3\cdots=(I-\beta\boldsymbol{A})^{-1}-I \tag{26}$$

其中，β 为权重衰减因子，通常为了保证数列的收敛性，β 的取值必须小于邻接矩阵 $\boldsymbol{A}$ 最大特征值的倒数.

5.3.5　基于局部路径的相似性计算(LP)

局部路径(Local Path)计算是在 CN 方法的基础上进一步考虑三阶邻居的贡献，其计算方法如式(27)所示：

$$S_{x,y}^{\mathrm{LP}}=A^2+\alpha A^3 \tag{27}$$

其中，α 为可调参数，用以控制三阶邻居的贡献强度，$\boldsymbol{A}$ 为网络的邻接矩阵，$(A^n)_{x,y}$ 为节点 (x,y) 之间路径长度为 n 的路径数.

5.3.6　基于平均通勤时间的相似度计算(ACT)

平均通勤时间(Average Commute Time，ACT)指的是一个随机粒子从节点 x 到达节点 y 平均需要走的步数 $m(x,y)$，此时，节点 x 和节点 y 的平均通勤时间定义为式(28)：

$$n(x,y)=m(x,y)+m(y,x) \tag{28}$$

其数值求解可以参考文献[11].

5.3.7　基于重启随机游走的相似度计算(RWR)

基于重启随机游走(Random walk restart)的相似度计算方法是目前连接预测的较好方法之一，在一些常见的网络数据集中都取得了最佳的 *AUC* 值[11]，该方法可看作是 PageRank 算法的拓展，被广泛的用于信息检索的各个领域[35]. 其假设是：在网络中进行随机游走时，粒子每走一步都会以一定的概率返回初始位置[42]. 假设粒子的返回概率为 $1-p$，$\boldsymbol{P}$ 为网络的马尔可夫概率转移矩阵，其元素 $P_{x,y}=a_{x,y}/k_x$ 表示节点 x 处的粒子下一步走到节点 y 的概率. 在网络中如果 x 和 y 相连，则 $a_{x,y}=1$；否则 $a_{x,y}=0$，而 k_x 则为节点 x 的度 $|E(x)|$.

如果某一初始时刻粒子在节点 x 处，则 $t+1$ 时刻粒子到达网络中各个节点的概率向量为

$$\boldsymbol{q}_x(t+1)=c\boldsymbol{P}^{\mathrm{T}}\boldsymbol{q}_x(t)+(1-c)e_x \tag{29}$$

式(29)中，e_x 为初始状态，由此可得稳定解为式(30)：

$$\boldsymbol{q}_x=(1-c)(I-c\boldsymbol{P}^{\mathrm{T}})^{-1}e_x \tag{30}$$

因此，元素 $q_{x,y}$ 表示从节点 x 出发的粒子最终以多少的概率走到节点 y，由此得到 RWR 的相似性计算方法如式(31)所示：

$$S_{x,y}^{\mathrm{RWR}}=q_{x,y}+q_{y,x} \tag{31}$$

5.4　有效性验证

5.4.1　实验环境

实验系统平台为 Mac OS X Yosemite 10.10.5，同时，为了提高对比的基准链路预测算法的矩阵计算效率，这部分程序采用 Matlab 2014b 实现，而 LsNet2Vec 模型则采用 Python 2.7 实现，同时启用 GPU 并行计算. 实验仪器的硬件性能为 2.5 GHz Intel i7 四核八线程处理器，内存为 16 GB，GPU 为 NVidia GeForce GT750 2GB.

5.4.2　*AUC* 对比

我们将 LsNet2Vec 模型在各个数据集上进行了 50 次的随机重复实验，通过 *AUC* 值，验证 LsNet2Vec

模型用以大规模网络链接预测问题的有效性. 如表 3 所示,为各个算法在所有数据集上重复实验所取得的平均值,其中 LsNet2Vec 的 *AUC* 值为在不同超参选择下的对应平均最优值.

表 3 各方法在各个数据集上的 *AUC* 值对比

Dataset	N2V	CN	AA	RA	KA. 01	LP. 0001	ACT	RWR
WVN	0.9523	0.9406	0.9424	0.9612	0.3802	**0.9761**	0.8977	0.9671
HCN	0.9852	0.9823	0.9832	0.9841	0.4786	0.9850	0.9660	**0.9880**
ACN	**0.9908**	0.9335	0.9349	0.9359	0.9741	0.9728	0.9409	0.9759
CCN	**0.9880**	0.9218	0.9229	0.9228	0.9699	0.9556	0.9241	0.9583
EEN	**0.9794**	0.8977	0.9055	0.9047	0.9578	0.9541	0.9421	0.9561
ESN	0.9220	0.6775	0.6994	0.6974	**0.9254**	0.8936	0.9040	0.9150
SSN	0.9033	0.5700	0.5744	0.5752	0.8993	0.8969	0.9004	**0.9441**
UEN	**0.9785**	0.5765	0.5764	0.5784	0.8895	0.6973	0.8265	0.8258
DCN	**0.9847**	0.8160	0.8187	0.8178	0.9696	0.9527	0.9586	0.9543
APN1	**0.9724**	0.8341	0.8348	0.8350	0.9182	0.9079	0.9052	0.9003
APN2	**0.9945**	0.7236	0.7459	0.7459	0.8444	0.8561	0.9230	0.9141
PRN	**0.9554**	0.5620	0.5617	0.5617	0.9130	0.7097	0.9072	0.9003
YSN	**0.8717**	0.6254	0.7035	0.7021	0.7509	0.8001	0.8559	0.8484
TRN	**0.9498**	0.5653	0.5654	0.5653	0.9017	0.6899	0.9016	0.8902
CRN	**0.9635**	0.5739	0.5736	0.5737	0.9122	0.7016	0.8998	0.9018
WTN	**0.8259**	0.5160	0.4948	0.5384	0.7985	0.8013	0.8125	0.8246

注:加粗波浪下划线为最大值,直下划线为次大.

值得注意的是,通过我们的初步实验,在大规模的真实数据集上,无论是对物理内存的需求还是算法的时间开销,现有基准对照方法根本无法直接在我们现有的实验硬件条件下进行相似性计算用以链接预测. 因此,除了 LsNet2Vec 模型是直接在上述大规模网络数据集的全集上进行进行训练和计算外,其他对照方法都只能采用社区抽样来完成评估. 为了有效和科学地获取现有基准方法在各个数据集上的表现,我们通过在每个大规模网络中分别按深度优先和广度优先各进行 100 次社团随机抽样,从而得到随机选取的若干小型社团(每个社团 10 000 个左右的节点);然后,在每个抽样得到的社团上进行 10 次重复实验;最后,计算每个算法在 $2\times100\times10$ 次实验上取均值,作为该方法在数据集上的表现.

由表 3 中的(3/4/5 列)实验结果可以发现,在以节点邻居的局部信息为参照的计算方法中,共同邻居类的相似性计算方法(CN/AA/RA)对于网络结构的密度或者平均集聚度系数特征要求较高,这三类算法的预测效果随着网络的密度或平均集聚系数的下降而变得非常不理想,例如 AA 算法在 WTN 数据集中 0.4948 的 *AUC* 值甚至还不如以 0.5000 为基准的随机猜测方法好. 然而,还应该注意到的是:在高密度的网络,如 CCN、ACN、HCN 以及 WVN 网络中,CN、AA 和 RA 算法也同样也获得了非常高的 *AUC* 值,这与文献[34]等结论一致. 同时,对于 CN、AA 以及 RA 的对比而言,资源分配算法(RA)相对于其他两种算法而言更好,原因在于 RA 算法的计算过程能够从根本上解释资源传递和节点之间的非线性关系,这种计算方式在节点的平均度较高的网络中效果更为明显,如 ACN 和 WVN 网络.

在基于路径信息的相似性计算方法中(6/7 列),KA(0.01)算法的表现最为突出,其在多个数据集上取得了次优于 LsNet2Vec 模型的结果,甚至在数据集 ESN 上取得了最优. 相对于 CN 类算法而言,KA 和 LP 算法由于引入了高阶邻居的贡献,使得原来很多在低阶环境下无法被区分的相似节点在高阶环境中变得可分,因此相对于直接的近邻算法(CN 类)而言其相似性的匹配性能普遍要好. 虽然,用过引入高阶邻居的贡献能够在一定程度上提升算法的预测效果,但同时这也带来了一个重要的负面影响,即 KA 算法的计算复杂度以及时间开销都是非常大的,而 LP 算法次之,因此从本质上来讲,这类算法并不适用于大规模网络中的链接预测或节点的相似度计算等问题.

在随机游走类的算法中(8/9 列),基于重启随机游走的相似性计算方法 RWR 表现最为突出,甚至在 HCN 和 SSN 数据集上取得了最优的成绩,同时取得了 3 个第二的预测效果. 此外,ACT 算法也取得了 3 个第二的对比效果. 但在实验中我们发现,虽然通过对数据集进行了社团采样,将数据量降为原来的 0.01 倍甚至 0.001 倍,RWR 和 ACT 算法相对于其他算法而言仍然较为耗时. 由于 RWR 和 ACT 算法的计算复杂度都是 $O(n^3)$,因此在大规模

的数据集中，其计算效率已变得十分低下.

通过对比发现，LsNet2Vec 虽然在 DCN、YSN 和 WTN 数据集上并没有十分明显的 *AUC* 提升，但由于现有的计算方法在这些数据集上的计算开销十分庞大. 从而使得从根本上来看，现有的方法在这些数据集上是无法进行不同节点之间的相似性计算的，因此在大规模网络结构中，现有的计算方法具有很大的局限性. 通过进一步分析发现：LsNet2Vec 模型在 UEN、APN1、APN2、PRN、TRN 和 CRN 数据集上相对现有算法而言有较大的性能提升，分别高于次优方法 8.9%、5.42%、7.15%、4.24%、4.81% 和 5.13%. 但是在 ESN，SSN 两个社会网络类的数据集以及两个小型数据集 WVN，HCN 中，LsNet2Vec 模型取得的效果并不十分理想. 特别是在 SSN 网络中，低于现有方法 4.08%，其原因是在 WVN，HCN 数据集中，不同节点之间的关系非常的紧密，网络的稀疏性不高，同时节点的平均度却非常高，使得基于多阶共现的 LsNet2Vec 模型并不能有效地识别不同节点之间的相似性与差异性. 而在 ESN，SSN 这样的社会关系网络数据集中，由于节点之间的不同阶影响差异较大，但是在 LsNet2Vec 模型中却被平等看待，因此效果稍弱. 但总的来说，相比于现有的链接预测方法，LsNet2Vec 模型在众多种类的数据集中都取得了不错的结果.

5.4.3　LsNet2Vec 性能解释

本节主要对 LsNet2Vec 模型取得相应效果的原因进行分析. 主要包括两个方面：

(1) 预测效果较好的原因. 如模型推理部分所述，LsNet2Vec 模型主要基于被预测节点的邻居节点的结构特征综合来表达自身的结构特征. 通过在网络中进行随机游走获得重构的序列化数据集后，针对特定窗口 N 大小的序列进行采样训练，因此对于一个节点而言，其结构特征理论上由其 N 阶邻居共同确定，这一点和 KA 方法以及 LP 方法有一定的相似性. 换句话说，一个节点的特征向量既由自身 N 阶邻居节点来估计，也被用于其 N 阶邻居节点特征向量的估计中.

(2) 计算复杂度相对较低的原因. 从本质上说：LsNet2Vec 模型可以看作是对原始网络节点结构特征的一种降维算法. 在特征的存储结构上，LsNet2Vec 模型和现有的网络计算中基于邻接矩阵的存储方式不同，LsNet2Vec 模型对节点的结构特征采用固定维度的向量进行存储，通常只有 20～1000 维. 通过训练得到节点结构特征向量后，能够大大降低后续应用的计算复杂度，例如计算节点相似度时，只用计算任意两个节点向量的余弦相似性即可. 同时，在 LsNet2Vec 模型的训练过程中，我们在每个节点上选取了较为简单的二分类，结合 Huffman 树进行存储后，利用随机梯度上升算法进行寻优，有效降低了计算的复杂度. 最后由于使用了在线的随机梯度上升算法，因此可以增量式地处理新增的节点数据，而不必重新训练，大大地节约了模型的训练时间.

5.5　超参讨论

LsNet2Vec 模型中包含了节点向量维度 *V-Size*、随机游走步长 *MAX_LENGTH*、随机游走次数 *WALK_TIMES* 以及预测窗口 *Window-Size N* 这 4 个超参. 本节将通过实验，探讨上述超参的选取对于 LsNet2Vec 模型用以链接预测问题性能好坏的影响方式，并给出 LsNet2Vec 模型在该问题中超参选取的建议. 在下文超参讨论中，我们采用控制变量的方法，取基准超参为 $MAX\_LENGTH=50$；$WALK\_TIMES=1.0\times Node\_number$；$V\text{-}Size=100$；$Window\text{-}Size=6$；当讨论某一个超参时，其他超参以上述基准超参为准，不进行变更.

5.5.1　节点特征向量维度

选取在初始探索性实验中 *AUC* 值因向量维度变动幅度较大的数据集进行进一步的对比实验. 将特征向量维度变更范围设置为[50，300]，同时每次增加 25 维进行一次 *AUC* 评测实验，实验结果如图 6 所示.

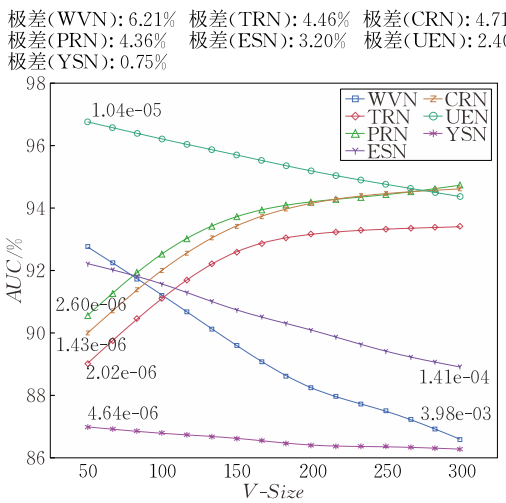


图 6　节点特征向量维度与 *AUC* 关系

通过对比实验发现，向量维度对网络链接预测的影响方式可以分为两种：对于网络密度较大的数据集，随着节点特征向量维度的增加，*AUC* 值逐步

下降，并且在同等变化区域内，密度越大，*AUC* 值下降越快. 对于网络密度较小的数据集，随着节点特征向量维度的增加，*AUC* 值逐步上升，同时，在同等变化区域内，网络密度越小，*AUC* 值上升越快.

因此，节点特征向量维度参数的选取需要和网络拓扑结构相适应. 对于密度较小的网络（一般这类网络通常也具备较大的节点数），其节点的结构特征分布较为分散，需要用更多的向量维度去获取不同分散特征（节点）之间的相似性与差异性. 而对于密度较大的网络，节点相对集中，如果采用过多的向量维度去衡量节点结构特征的相似性，那么则会减少具有重要区分度的特征权重，从而给链接预测带来不准确性.

5.5.2 序列化随机游走步长与随机游走次数

随机游走步长 *MAX_LENGTH* 和预测窗口 *Window-Size N* 之间存在一个基本范围限制，通常需要满足

$$MAX\_LENGTH \geq Window\text{-}Size,$$

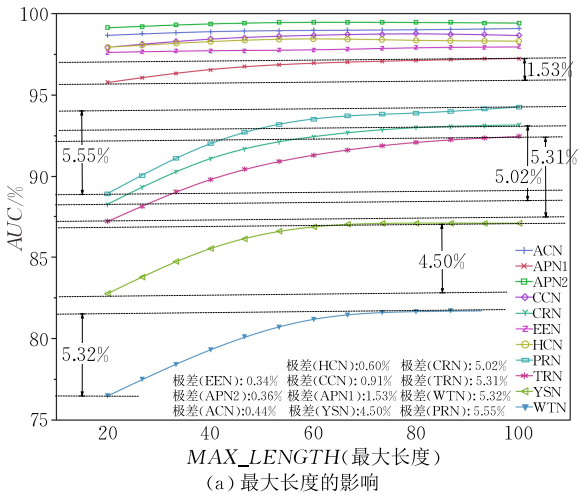
否则，在进行序列采样训练时，无法满足采样要求. 因此我们选取 *MAX_LENGTH* 变化区间为[20,100]，以 10 为步长进行递增，重复实验得到稳定结果后如图 7 所示.

从图 7(a)中可以发现，*MAX_LENGTH* 的变更对数据集链接预测 *AUC* 值的影响幅度（极差）是随着网络节点平均度/网络密度的降低而增加：当网络较为稀疏时，通过增加随机游走的步长能够有效地提升系统的预测性能. 当网络相对密集时，通过增加随机游走步长，并不能十分明显的提升预测效率. 同时，我们发现，对于固定的预测窗口 *Window-Size* = *N* 的参数而言，通过变更随机游走步长，*AUC* 极差的跳跃点发生在平均度约为 *N*/2 的数据集处. 如图 7(b)所示，当数据集的平均度小于 *N*/2=3 的时候，通过在区间[20,100]变更步长所产生的极差首先迅速下降，然后趋于平稳. 同样，网络密度对极差的影响也是相似的，如图 7(c)所示.

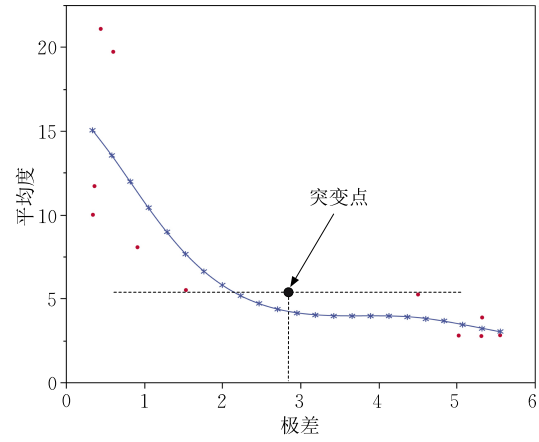
因此，随机游走的步长选取应结合网络的平均度（*Ave_degree*）以及预测窗口 *Window-Size* = *N* 来进行设定：

当 $Ave\_degree \approx N/2$ 时，可以通过大范围的实验来调整随机游走步长来获取较大的 *AUC* 变更信息.

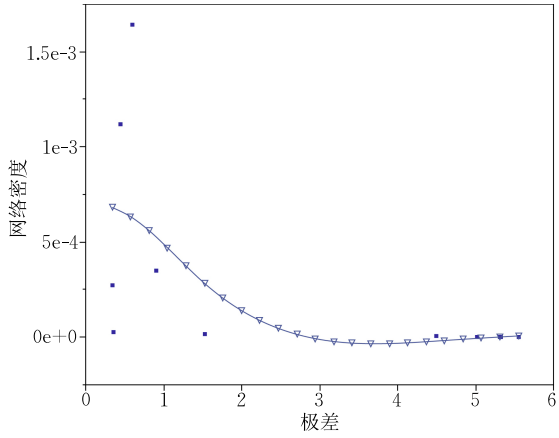
当 $Ave\_degree \ll N/2$ 或 $Ave\_degree \geq N/2$ 时，建议选取和网络直径相似的随机游走步长来进行小范围的调整实验，因为此时随机游走步长的调整并不会对 *AUC* 值造成太大的提升，如果再进行大范围的实验，会增加计算开销与复杂度.



(a) 最大长度的影响



(b) 极差变化



(c) 网络密度对极差的影响

图 7 随机游走步长与 *AUC* 关系

随机游走次数 *WALK_TIMES* 主要影响模型训练的充分性以及时间开销. 从本质来说，*WALK_TIMES* 和 *MAX_LENGTH* 存在一个均衡关系，当 *MAX_LENGTH* 增加时，可以通过适当减少 *WALK_TIMES* 来保证 *AUC* 不变的同时减少训练

的复杂度. 同样，当 *MAX_LENGTH* 固定时，可以通过增加 *WALK_TIMES* 来提升一定的 *AUC* 值，一般的：当 *WALK_TIMES* 大于网络中的边数时，这种提升并不明显，限于篇幅，关于 *WALK_TIMES* 的影响不再详细讨论.

5.5.3　预测窗口(*Window-Size N*)

预测窗口反映了被预测节点的结构环境信息，当预测窗口为 *N* 时，其环境邻居最多可达 *N* 阶. 通过将预测窗口控制在[4,11]，按照步长 1 进行重复实验，得到如图 8 所示关系. 图中实验数据集几乎都是先上升后下降，同时出现了相应的极值点.

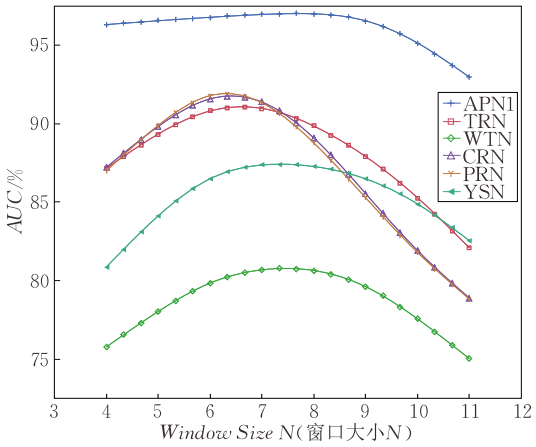


图 8　预测窗口与 *AUC* 关系

对于 LsNet2Vec 模型而言，扩大环境窗口并不会增加太多的计算量(固定维度的向量加减运算). 因此，理论上利用的全局信息越多，越有利于模型的预测. 但在本文 LsNet2Vec 模型中，我们采用的是对窗口范围内的环境向量直接加和的形式(式(7))获得环境特征向量，即不存在相对位置差异，也不存在权重的变化，从而导致一阶邻居和二阶邻居和 *N* 阶邻居的贡献相同. 当窗口较小时，这些邻居的结构贡献对于预测节点是相似的，增加到二阶，三阶等将有利于提供更多的信息. 但当预测窗口超出一定范围，就会使得本来的一阶邻居和 *N* 阶邻居的贡献大不相同的情况在 LsNet2Vec 模型中成为相似的(这是不合理的)，因此反而使得 LsNet2Vec 模型的预测准确性下降. 针对这个问题，我们将在后续的研究中进一步改进：通过引入环境向量的贡献衰减函数，针对不阶的邻居给予不同的结构贡献权重，进一步优化预测模型.

6　结论与下一步研究

网络链接预测是复杂网络分析和数据挖掘领域的热点问题之一. 现有的链接预测方法主要针对特定的网络结构进行详细的分析和设计，从而使得预测方法变得较为复杂，因此难以适应越来越庞大的网络结构. 本文针对大规模网络链接预测问题中存在的网络特征稀疏性和计算复杂度高的情况，提出了基于网络结构特征降维的 LsNet2Vec 模型.

首先，通过对网络中节点特征的概率拟合，以节点的环境特征为输入，节点自身特征为输出，无监督学习得到了网络中节点特征的分布式表达. 然后，通过学习得到的固定低维度特征向量就可以直接而又快速地计算出大规模网络中任意节点之间的相似度. 最后，基于大规模真实数据集的实验表明，LsNet2Vec 模型在大规模网络上具备可计算性以及相对较优的预测性能，并在多个数据集上对以往的链接预测方法有了很大的提升.

当然，LsNet2Vec 模型也存在一定的不足，主要表现在以下两点：(1) LsNet2Vec 模型在节点环境向量的构建过程中并没有考虑不同阶邻居所带来的不同影响，而是将它们统一看待，削弱了近邻的影响；(2) LsNet2Vec 模型对于特殊网络(例如，高密度或集聚度网络)的适应性不足，如在 WTN 网络中预测的 *AUC* 值只有 82.56%.

后续的可能研究思路主要包括研究节点特征环境的不同构造方法，如增加权重衰减等；研究不同类型网络的节点结构(或边结构)特征的分布式表达问题，例如带权网络节点/边结构特征的分布式表达，多重网络的节点结构特征的分布式表达等；研究不同网络(异质网络)结构特征的统一表达问题，以此探索跨平台网络中不同节点结构特征的相似性匹配问题.

参　考　文　献

[1] Zhou T, Lu L, Zhang Y-C. Predicting missing links via local information. The European Physical Journal B, 2009, 71(4): 623-630

[2] Narang K, Lerman K, Kumaraguru P. Network flows and the link prediction problem//Proceedings of the 7th Workshop on Social Network Mining and Analysis. New York, USA, 2013: 1-8

[3] Lei C, Ruan J. A novel link prediction algorithm for reconstructing protein-protein interaction networks by topological similarity. Bioinformatics, 2013, 29(3): 355-364

[4] Du N, Gao J, Zhang A, et al. De-noise biological network from heterogeneous sources via link propagation//Proceedings

of the IEEE International Conference on Bioinformatics and Biomedicine. New York, USA, 2012: 1-6

[5] Brandão M A, Moro M M, Lopes G R, et al. Using link semantics to recommend collaborations in academic social networks//Proceedings of the 22nd International Conference on World Wide Web. Rio de janeiro, Brazil, 2013: 833-840

[6] Yu Q, Long C, Lv Y, et al. Predicting co-author relationship in medical co-authorship networks. PLoS ONE, 2014, 9(7): e101214

[7] Liben-Nowell D, Kleinberg J M. The link prediction problem for social networks. Journal of the American Society for Information Science and Technology, 2007, 58(7): 1019-1031

[8] Tylenda T, Angelova R, Bedathur S. Towards time-aware link prediction in evolving social networks//Proceedings of the 3rd Workshop on Social Network Mining and Analysis. New York, USA, 2009: 1-10

[9] Guns R, Rousseau R. Recommending research collaborations using link prediction and random forest classifiers. Scientometrics, 2014, 101(2): 1461-1473

[10] Fire M, Tenenboim L, Lesser O, et al. Link prediction in social networks using computationally efficient topological features//Proceedings of the IEEE Third International Conference on Social Computing and 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust. Boston, USA, 2011: 73-80

[11] Lü Lin-Yuan. Link prediction on complex networks. Journal of University of Electronic Science and Technology of China, 2010, 39(5): 654-661(in Chinese)
(吕琳媛. 复杂网络链路预测. 电子科技大学学报, 2010, 39(5): 654-661)

[12] Lü L, Zhou T. Link prediction in complex networks: A survey. Physica A: Statistical Mechanics and Its Applications, 2011, 390(6): 1150-1170

[13] Zhang J, Yu P S, Zhou Z-H. Meta-path based multi-network collective link prediction//Proceedings of the 20th International Conference on Knowledge Discovery and Data Mining. New York, USA, 2014: 1286-1295

[14] Thi D B, Ichise R, Le B. Link prediction in social networks based on local weighted paths//Dang T K, Wagner R, Neuhold E et al, eds. Future Data and Security Engineering. Switzerland: Springer International Publishing, 2014: 151-163

[15] Liu W, Lü L. Link prediction based on local random walk. Europhysics Letters, 2010, 89(5): 58007

[16] Jin T, Xu T, Chen E, et al. Random walk with pre-filtering for social link prediction//Proceedings of the 9th International Conference on Computational Intelligence and Security. Leshan, China, 2013: 139-143

[17] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521(7553): 436-444

[18] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model. Journal of Machine Learning Research, 2003, 3: 1137-1155

[19] Mikolov T, Zweig G. Context dependent recurrent neural network language model//Proceedings of the IEEE Workshop on Spoken Language Technologies. Miami, USA, 2012: 234-239

[20] Huang E H, Socher R, Manning C D, et al. Improving word representations via global context and multiple word prototypes //Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Pittsburgh, USA, 2012: 873-882

[21] Socher R, Perelygin A, Wu J Y, et al. Recursive deep models for semantic compositionality over a sentiment treebank// Proceedings of the Conference on Empirical Methods in Natural Language Processing. Seattle, USA, 2013: 1631-1642

[22] Roth M, Woodsend K. Composition of word representations improves semantic role labeling//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Doha, Qatar, 2014: 407-413

[23] Bengio Y, Schwenk H, Senécal J-S, et al. Neural probabilistic language models//Holmes D E, Jain L C eds. Innovations in Machine Learning. Springer Berlin Heidelberg, 2006: 137-186

[24] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space. arXiv preprint, 2013, cs.CL: 1-12

[25] Liben-Nowell D, Kleinberg J M. The link-prediction problem for social networks. Journal of the American Society for Information Science and Technology, 2007, 58(7): 1019-1031

[26] Liu D, Wang Y, Jia Y, et al. LSDH: A hashing approach for large-scale link prediction in microblogs//Proceedings of the 28th AAAI Conference on Artificial Intelligence. Québec, Canada, 2014: 3120-3121

[27] Lü L, Jin C-H, Zhou T. Similarity index based on local paths for link prediction of complex networks. Physical Review E, 2009, 80(4): 046122

[28] Lichtenwalter R N, Lussier J T, Chawla N V. New perspectives and methods in link prediction//Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining. New York, USA, 2010: 243-252

[29] Huang Li-Wei, Li De-Yi, Ma Yu-Tao, et al. A meta path-based link prediction model for heterogeneous information networks. Chinese Journal of Computers, 2014, 37(4): 848-858(in Chinese)
(黄立威, 李德毅, 马于涛等. 一种基于元路径的异质信息网络链路预测模型. 计算机学报, 2014, 37(4): 848-858)

[30] Liu Ye, Zhu Wei-Heng, Pan Yan, et al. Multiple sources fusion for link prediction via low-rank and sparse matrix decomposition. Journal of Computer Research and Development, 2015, 52(2): 423-436(in Chinese)

(刘冶，朱蔚恒，潘炎等. 基于低秩和稀疏矩阵分解的多源融合链接预测算法. 计算机研究与发展，2015，52(2)：423-436)

[31] Wu Zu-Feng, Liang Qi, Liu Qiao, et al. Modified link prediction algorithm based on AdaBoost. Journal on Communications, 2014, 35(3): 116-123(in Chinese)
(吴祖峰，梁棋，刘峤等. 基于 AdaBoost 的链路预测优化算法. 通信学报，2014，35(3)：116-123)

[32] Li Yu-Hua, Xiao Hai-Ling, Li Dong-Cai, et al. Research of dynamic link prediction method based on link importance. Journal of Computer Research and Development, 2011, 48(S3): 40-46(in Chinese)
(李玉华，肖海岭，李栋才等. 基于链接重要性的动态链接预测方法研究. 计算机研究与发展，2011，48(S3)：40-46)

[33] Pujari M, Kanawati R. Link prediction in multiplex networks. Networks and Heterogeneous Media, 2015, 10(1): 17-35

[34] He Y L, Liu J N K, Hu Y X, et al. OWA operator based link prediction ensemble for social network. Expert Systems with Applications, 2015, 42(1): 21-50

[35] Lv B, Yu W, Wang L, et al. Efficient processing node proximity via random walk with restart//Proceedings of the 16th Asia-Pacific Web Conference. Changsha, China, 2014: 542-549

[36] Ogata H, Suzumura T. Towards scalable X10 based link prediction for large scale social networks//Proceedings of the 23rd International Conference on World Wide Web. Seoul, Korea, 2014: 1327-1332

[37] Shin D, Si S, Dhillon I S. Multi-scale link prediction//Proceedings of the 21st ACM International Conference on Information and Knowledge Management. New York, USA, 2012: 215-224

[38] Richard E, Gaiffas S, Vayatis N. Link prediction in graphs with autoregressive features. Journal of Machine Learning Research, 2014, 15(1): 565-593

[39] Chen Z, Chen M, Weinberger K Q, et al. Marginalized de-noising for link prediction and multi-label learning//Proceedings of the 29th AAAI Conference on Artificial Intelligence. Austin, USA, 2015: 1707-1713

[40] Lorrain F, White H C. Structural equivalence of individuals in social networks. The Journal of Mathematical Sociology, 1971, 1(1): 49-80

[41] Adamic L A, Adar E. Friends and neighbors on the Web. Social Networks, 2003, 25(3): 211-230

[42] Fouss F, Pirotte A, Renders J-M, et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(3): 355-369

LI Zhi-Yu, born in 1991, Ph. D. candidate. His research interests include social computing, machine learning.

LIANG Xun, born in 1965, Ph. D., professor, Ph. D. supervisor. His research interests include neural networks, support vector machine and social computing.

ZHOU Xiao-Ping, born in 1985, Ph. D. candidate. His research interests include Web data mining and social computing.

ZHANG Hai-Yan, born in 1975, Ph. D. candidate. Her research interests include complex networks, social computing and recommend systems.

MA Yue-Feng, born in 1976, Ph. D. candidate. His research interests include data mining, machine learning and pattern recognition.

Background

Since complex networks are likely to play an essential role in data mining research areas, which make the problem of link prediction in complex networks also attracted much attention recently. The problem of link prediction can be categorized into two classes, namely, missing links prediction and future links prediction. Missing links prediction is the prediction of unknown links in sampling networks; and the other is the prediction of links that may exist in the future of evolving complex networks. Both missing and future links prediction are considered important subtasks in link prediction.

Until now, most of the methods for link prediction are designed based on the assumption of node similarity, which defined by using the essential features of nodes. Those features can be structural or contextual, which means that two nodes are considered to be similar if they have many common features.

The objective of link prediction is to estimate the likelihood that a link exists between two nodes, making the sparsity and huge size of networks become two of the main challenges remain in link prediction problems. Although there are many similarity-based algorithms, such as Common Neighbor (CN) algorithm, Katz algorithm, Local Path (LP)

algorithm, Random Walk Restart (RWR) algorithm etc., which have been proposed to handle this essential problem in the small complex networks, the empirical observations show that the stability and usability in large-scale networks of existing algorithms is usually very low, which means, for a large network with millions of nodes, this number can easily double or triple, making learning and predicting of unknown links very expensive.

In this work, we describe here a new approach to predict unknown links in large-scale networks according to the unsupervised machine learning. The main idea of our method is mapping the features of node in large-scale networks into a lower and fixed dimension of vector in the set of real numbers.

We conduct extensive experimental analysis on sixteen famous datasets, which provided Stanford Network Analysis Project. Then, we present a controlled comparison of the LsNet2Vec model against several strong baseline link prediction methods on a fixed dataset, with *AUC* testing. Result showed that the LsNet2Vec model performs comparably with state-of-the-art methods, and consistently outperforms models, such as Katz and RWR etc., in various experiment settings.

This work is partly supported by the National Natural Science Foundation of China (Grant Nos. 71271211 and 71531012), the Natural Science Foundation of Beijing (Grant No. 4132067), the Fundamental Research Funds for the Central Universities (the Research Funds of Renmin University of China, Grant No. 10XNI029), the Outstanding Innovative Talents Cultivation Funded Programs 2015 of Renmin University of China.