DNPS:基于阻尼采样的大规模动态社会网络结构特征表示学习

李志宇[1)] 梁 循[1)] 徐志明[1)] 齐金山[1)] 陈燕方[2)]

[1)](中国人民大学信息学院 北京 100872)

[2)](中国人民大学信息资源管理学院 北京 100872)

摘 要 网络特征表示学习通过对网络节点之间的关系(结构或属性)进行分析,得出网络特征的低维度表达.现有的针对网络特征学习的方法多基于静态和小规模的假设(如静态的语言网络),并没有针对社会网络的特有属性进行修正学习,因此,现有的学习方法无法适应当前社会网络所具备的动态性、大规模甚至超大规模等特性.该文在已有研究基础上,提出了基于动态阻尼正负采样的社会网络结构特征嵌入模型(Damping Based Negative-Positive Sampling of Social Network Embedding,DNPS).通过对不同阶层的网络节点关系进行正负阻尼采样,同时构建针对新增节点的动态特征学习方法,使得模型对于大规模社会网络在动态变化过程中的结构特征的提取变得可行,以此获得的节点特征表达具备更好的动态鲁棒性.通过选取3个大规模的动态社会网络的真实数据集和在社会网络的动态链接预测问题的实验中发现:DNPS相对于基准模型(DeepWalk/LINE)在预测准确率以及时间效率上都取得了较大的性能提升.同时,DNPS的学习结果还可以被应用于社会网络的相关研究子领域.例如,在大规模以及动态性的环境下,研究大规模动态社区发现、社会网络用户推荐、标记分类等问题.

关键词 社会网络;节点嵌入;节点特征;神经网络;链接预测;社会媒体

中图法分类号 TP311 **DOI号** 10.11897/SP.J.1016.2017.00805

DNPS: Damping Based Negative-Positive Sampling of Large-Scale Dynamic Social Network Embedding

LI Zhi-Yu[1)] LIANG Xun[1)] XU Zhi-Ming[1)] QI Jin-Shan[1)] CHEN Yan-Fang[2)]

[1)](*School of Information, Renmin University of China, Beijing* 100872)

[2)](*School of Information Resource Management, Renmin University of China, Beijing* 100872)

Abstract Network feature learning can obtain the low-dimensional representations of network by analyzing the relationships (structures or attributes) between nodes. However, there are many nodes embedding methods, which based on assumptions of static and small-scale (such as language networks), are unable to adapt to social networks, because social networks have their specific properties such as dynamic and large-scale. Based on current researches, this paper propose a damping based positive and negative sampling model for learning nodes embedding of social networks. By sampling nodes at different levels with damping, at the same time design an incremental learning method for newly added nodes, which makes it possible for learning nodes features extracting during the dynamic changing process, thus to learning a better representations of social networks. Finally, we select three large-scale, dynamic and real-life social networks for dynamic link prediction task. The results show that, compared with DeepWalk and LINE methods,

收稿日期:2016-05-30;在线出版日期:2016-09-28.本课题得到国家自然科学基金(71271211,71531012)、北京市自然科学基金(4172032)、中国人民大学科学研究基金(10XNI029)及中国人民大学2016年度拔尖创新人才培育资助计划成果资助.**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.**徐志明**,男,1993年生,硕士研究生,主要研究方向为社会网络、并行计算.**齐金山**,男,1978年生,博士研究生,主要研究方向为社会网络、身份识别.**陈燕方**,女,1992年生,博士研究生,主要研究方向为社会网络、数据挖掘.

DNPS have achieved greater performance in prediction accuracy and time efficiency. The learned node vectors by DNPS model can be used in many subfields of social network research. For example, we can use it for large-scale dynamic social community discovery, user recommendation, and user labeling.

Keywords social network; nodes embedding; nodes features; neural network; link prediction; social media

1 引 言

社会媒体网络(Social Media Networks,简称社会网络)是复杂信息网络的主要代表之一,它可以看作由节点(Nodes)和边(Edges)所构成的复杂抽象组合,其中节点之间通过边的有向(Directed)或无向(Undirected)的链接进行信息与功能的交互.随着社交媒体的广泛流行,社会网络所包含的数据量及其种类急剧增长,其中,社会网络的动态性(Dynamic of Social Networks)则是当前大规模社会网络发展过程中的重要特征.由此而引发的包括网络维数灾难、计算复杂性等问题已成为社会网络研究领域亟待解决的一个重要问题[1-3].

以新浪微博为例,在节点数量上,截至 2015 年 9 月 30 日,微博月活跃用户为 2.22 亿,相比 2014 年同期增长 33%.在节点关系上,微博已从原有的关注关系进一步衍生出营销关系、转发关系、话题关系与行业关系等多重关系网络的组合.在节点与边的属性内容上,用户所发布和参与的形式更加多样,既包括传统的文本、图片与视频形式,还衍生出微博投票、微博众筹、微博旅游等多种互动参与形式①.正是由于这种大规模社会网络原始表达的多样性和高量级,使得对于社会网络的表示学习与分析的需求急剧增长.

表示学习(Representation Learning)亦称特征学习(Feature Learning),是当前机器学习研究领域的一个重要研究方向[4].表示学习的基本目的是:通过对数据进行表达形式的变化,使其数据所包含的信息更加容易被提取和分析[4],即将原来由人工设定的特征工程转换为机器的自我学习过程.例如,在文本表示学习中,词向量通过对文本的共现情景进行的自动学习,得到词语在低维度(Low Dimensions)上的表示[5-7].通过研究发现,这种低维度的表示向量能够有效地显示出词语之间的语义关系[7],而且更易于被应用到其它的系统中去.同样,社会网络表示学习则是通过对社会网络的结构特征和内容特征等进行学习,得到网络的不同表现形式,并以此为输入,将其拓展到例如推荐系统或隐私保护系统等衍生应用中去.如图 1 所示,一个较好的网络表示学习方法能够有效地学习得到原有网络中节点或者社区之间的分布关系(图 1 中的核心社区之间存在三角距离关系).这些分布关系体现在原有网络拓扑结构中的各种离散属性关系能够有效地表现

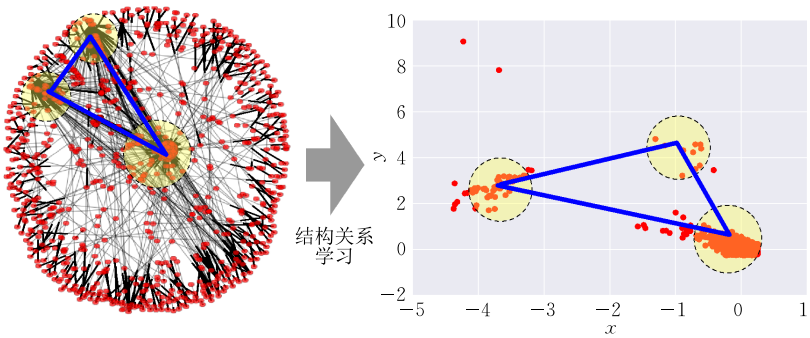


图 1 网络结构特征表示学习

在连续的向量空间中.

社会网络的表示学习相比于其它数据类型的表示学习更为复杂.从数据构型上看,社会网络的表示学习涉及到了实体关系(离散关系,连续关系)学习与实体属性(节点属性,边属性)学习,以及它们之间的交互学习.从数据体量上看,社会网络的表示学习涉及到了从 10^2～10^9 甚至更大的节点或边集规模.其中,每一个节点既可以被单独的划分成为一个类,

① 2015 年微博用户发展报告. http://www.199it.com/archives/422583.html. 2016, 2, 17

也可以和众多的节点按照社区标签的形式划分成多个类.

然而，随着大规模社会网络的发展，以往的基于概率生成模型和谱模型等传统网络表示框架已经无法有效地适应现代大规模社会网络的动态性和海量性等特征. 因此，针对大规模社会网络设计有效的网络表示模型已经逐步成为当前的研究热点之一. 近年来，深度学习的各类算法在图像识别，语音处理以及文本分析等多个领域取得了较大的突破[8]. 同时也有学者开始注意到深度学习算法在网络特征学习上应用的可能性[9-12]. 例如：Perozzi 等人[9]提出的 DeepWalk 模型，通过结合随机游走的方式，参考 Mikolov 等人[7]的 Skip-gram 文本词向量学习方式，给出了对应的网络特征分布式表示方法. 该模型在一定程度上解决了网络训练数据的稀疏性问题，在数据量较少的情况下，以部分训练数据达到传统方法在全集训练数据上才能得到的训练效果. 此后，包括 LINE[12]，GraRep[10]等方法相继被提出，我们将在第 2 节对它们进行对比分析.

本文在已有关于网络节点特征学习模型的基础上[13]，提出了基于阻尼的动态正负边集采样的网络表示学习算法，其核心为：针对社会网络的层次性特征，构建基于有阻尼的节点共现学习，同时对于网络的动态变化设计基于局部搜索的增量式采样算法，通过融合节点特征学习模型以及社会网络自身特性，设计具备动态以及大规模适应性的社会网络表示学习模型.

主要贡献. 首先，我们提出了基于阻尼采样的网络特征学习模型 DNPS，有效的解决了大规模社会网络的动态适应性问题；其次，通过各项对比实验表明，DNPS 在算法的准确度以及时间效率上优于基准的对比算法，取得了较大的性能提升；最后，我们通过建立项目网站①，开源了项目程序，为本文算法的后续对比以及在其它领域的衍生应用提供相应的参考.

本文第 2 节对网络特征学习领域涉及到的主要算法和模型进行相应的对比综述；第 3 节则对文中模型所涉及的核心概念与核心问题进行形式化的定义和规范化的表达；第 4 节则对基于阻尼采样的动态网络学习模型进行分析与推导；第 5 节为实验设置与结果分析部分，主要分析基准方法与 DNPS 模型在动态链接预测问题上的准确性以及 DNPS 对社会网络动态变化的适应能力；最后，本文第 6 节给出文章的主要结论以及未来的可能研究方向.

2　相关工作

网络表示学习目标是通过节点关系学习，将原有的高维度离散特征关系转换为低维度的连续特征表示，这种低维的表示形式能够有效地克服原有网络存在的稀疏性问题，并且可以进一步作为其它应用的输入，对原始网络构建拓展的分析模型. 传统的网络表示学习方法往往是通过对网络的邻接矩阵(Adjacency matrix)或关联矩阵(Incidence matrix)采用降维分析的方法，通过求解特征向量的形式来获取网络的低维表达. 常见的方法有 IsoMap[14]，LLE[15]，Laplacian Eigenmap[16]，LLC[17]等.

然而，由于这些常见方法通常是基于小规模网络，并且不适应动态网络的变化，因此在节点数目较多的情况下，传统的算法的计算复杂度往往是无法承受的. 受到深度学习在文本网络应用的启发，近两年来有针对较大规模的网络特征分布式特征表示学习的算法被提出. 其中，具有代表性的算法是 DeepWalk[9]，LINE[12]以及 GraRep[10]算法.

如表 1 所示，为当前最新的基于网络结构学习的表示模型. 上述 3 个模型都可以看做是在一定程度上受到 Mikolov 等人[7]的 Word2vec 模型的启发而得到，但是在模型的表达能力以及应用范围上存在一定的区别. 接下来本文将对表 1 中 3 个模型进行一定的分析和对比.

表 1　基于结构的网络表示学习模型

模型	核心算法与观点	表示对象	类型*	实验规模	实验环境	评测数据集
DeepWalk[9](2014)	＞随机游走产生输入序列 ＞基于 Skip-gram 模型	Node	UD/NW	1138499	单机，24 Cores @2.0GHz CPU，128GB 内存	社会网络
LINE[12](2015)	＞重构目标函数 ＞带权边采样算法	Node	D/UD W/NW	1985098	单机，40 Core @2.0GHz CPU，1TB 内存	文本/社交/引用网络
GraRep[10](2015)	＞学习网络的全局特征 ＞优化部分使用 SVD	Node	W/NW UD	10312	单机，4 Cores @3.4GHz CPU，16GB 内存	文本/社交/引用网络

注：类型*：D/UD 为有向/无向图；W/NW 为带权/非带权图；实验规模为模型所采用实验数据集的最大节点数.

① http://www.n2v.org/cjc_2016li.html

如图 2 所示，为对比模型的引用及其应用测试的关系图. 其中社会网络为 3 个模型共同对比应用，但是通过对模型的分析，可以发现 DeepWalk 模型，LINE 模型以及 GraRep 模型都未对社会网络做出特定的优化，例如增加与社会网络特征相适应的模块等等. 在图 2 所示的 3 个模型中，DeepWalk 首先被提出. DeepWalk 参照 Word2Vec 的训练模型，首先基于原始网络结构进行随机游走，产生 Word2Vec 的训练模型中所需要的序列化数据集，通过 Skip-gram 子模型得到网络节点的分布式表达. 通过模型在社会网络上的实验对比发现，相比于基准方法，DeepWalk 模型有效解决了网络训练数据的稀疏性问题，使其在较少的数据情况下超过全集数据模型所取得的训练效果. DeepWalk 模型的提出为网络特征结构的分布式表达提出了一个新的可行方向.

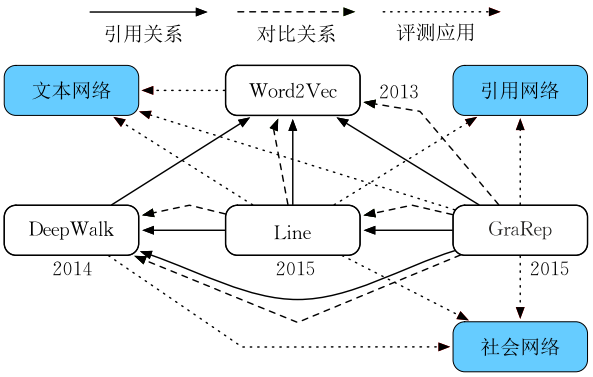


图 2 主要模型的引用对比关系

在此之后 LINE 模型被提出. LINE 模型将 DeepWalk 模型对节点关系的刻画从一阶拓展到了二阶. 通过对不同阶关系的刻画，设置不同的目标函数. 最后将一二阶关系得到的分布式表达进行拼接，从而获得节点的统一表达. 与 DeepWalk 不同的是，LINE 对于一阶二阶关系的表示训练过程都采用了基于边集负采样的方式进行，同时在文本网络、引用网络和社会网络中进行了实验. 相比于 LINE 提出的对一阶和二阶的特征获取，GraRep 模型则将 LINE 继续拓展了 k 阶关系的建模，通过进一步对 k 阶关系的逐一建模，从而获取网络节点从局部到全局的特征表示. 相似的 GraRep 模型仍然采用向量拼接的形式构建最后的统一表达. 然而，值得注意的是 GraRep 相比于前两者模型而言，其计算的复杂度较高，并不适用于大规模网络，因此本文不将 GraRep 方法纳入对比模型. 在上述的研究基础之上，本文提出了基于阻尼采样的动态社会网络分布式表达模型，相比 DeepWalk 模型，LINE 模型以及 GraRep 模型，本文提出的 DNPS 模型主要有以下特点与不同：

首先，本文针对大规模动态社会网络的特征提出了基于阻尼(Damping)的节点环境贡献，即认为处在不同阶层之间的节点其贡献强度是存在差异的，因此在进行共现学习时，其共现强度需要进行动态的调整.

其次，本文首次将社会网络的动态性问题考察引入到网络节点结构分布式特征的学习过程中来. 基于局部搜索的动态采样算法的引入在进一步丰富节点关系特征表达的同时，使得 DNPS 模型能够有效的对社会网络的动态变化过程中的结构特征进行学习和表示.

最后，相比于 DeepWalk 以及 LINE 模型，本文构建了友好的项目网站和开源程序，为相关领域(例如社会网络分析等)应用给出了良好的使用说明.

3 关键定义与基础

本节将对模型中所涉及到的关键概念进行形式化的描述与分析，同时给出了动态链接预测问题的定义与问题要求. 下面，首先给出文中涉及的符号含义及其表示，如表 2 所示.

表 2 核心符号说明

符号	含义
A,B,C,…	网络节点，正体大写
$\boldsymbol{v}_A,\boldsymbol{v}_B,\boldsymbol{v}_C,\cdots$	网络节点对应向量 $\boldsymbol{v}\in\mathbb{R}^d$
V_{t_i},E_{t_i}	网络在 t_i 时刻对应的节点集合与边集合
ω_{BA}	有向边 $B\to A$ 对应的权重，由输入源决定
δ_{AD}	可达路径中一次预测对应的动态阻尼系数
ε	全局阻尼系数，超参数
λ	采样增长覆盖率，超参数
d	向量维度数，超参数
$\lvert N_x\rvert$	负采样节点数，超参数

3.1 关键定义

定义 1. 动态社会网络是指网络的结构关系随着时间的推移而发生边的出现/消失与节点的出现/消失的现象的网络.

一般的，对于网络 $G_{t_i}=(V_{t_i},E_{t_i})$，在现有关于动态社会网络的研究中，都存在式(1)假设[18]，

$$V_{t_i}\subseteq V_{t_{i+1}},\ E_{t_i}\subseteq E_{t_{i+1}} \tag{1}$$

即随着网络的不断变化，在时间 t_{i+1} 时刻，网络的节点与边集合包含了时间 t_i 时刻的节点与边集合. 然而这种假设在一定程度上忽略了网络在动态变化的过程中还存在节点的消失以及边消失的情

况. 在本文中，我们将这种节点与边的消失同时纳入动态社会网络的表示学习中，设计基于增量学习边采样算法来适应网络的动态变化. 定义如下：

$$\begin{cases}\text{Appear}=\begin{cases}V_{t+1}^{AP}=V_{t+1}-V_{t+1}\cap V_t\\ E_{t+1}^{AP}=E_{t+1}-E_{t+1}\cap E_t\end{cases}\\ \text{Vanish}=\begin{cases}V_{t+1}^{VA}=V_t-V_{t+1}\cap V_t\\ E_{t+1}^{VA}=E_t-E_{t+1}\cap E_t\end{cases}\end{cases}\tag{2}$$

式(2)中：V_{t+1}^{AP}，E_{t+1}^{AP}分别为 $t+1$ 时刻相对于 t 时刻的网络中新增节点和边的集合，而 V_{t+1}^{VA}，E_{t+1}^{VA}则分别为网络中 $t+1$ 时刻相对于 t 时刻的消失节点和边的集合.

定义 2.　节点特征的分布式表达又可称为节点的特征向量，用以存储节点的结构特征.

传统方法对于节点的特征定义多为离散的，例如，节点的度与节点的中心性等特征定义. 本文则将节点特征定义其包含于一个连续的向量 $\boldsymbol{v}\in\mathbb{R}^d$，其中 d 为向量的维度数，通常 $d\ll|V|$，即特征向量的维度数要远远小于网络的节点数目. 为了更好的对网络进行抽象与模拟，本文将网络定义为带权有向图. 如图 3 所示，$\boldsymbol{v}_A$ 和 $\boldsymbol{v}_B$ 分别表示对应的节点向量，而 ω_{BA}，ω_{ABA} 则为两条边对应的权重系数.

图 3　节点向量的符号定义

如果网络为无向图，通常需要将原有无向图的每条边赋予双向边，同时在网络的处理(读取与存储)时做出针对性的修改和优化. 同样的，针对非带权网络，将权重默认为 1.0.

定义 3.　节点特征环境(Nodes Features of Environment，NFE)指的是在模型预测过程中，通过某种组合之后，用以代表被预测节点的环境构成.

NFE 表示为利用环境节点对节点特征本身的正确预测的概率组合，因此本文中，节点特征环境的选取过程又被称为正采样(Positive Sampling，PS). 节点特征环境的构建，对于网络特征的分布式学习起到非常重要的作用. 例如，在 DeepWalk[9] 模型中，作者仅仅考虑了节点之间的一阶相似性，这种一阶相似性可以看做广度特征的一个子集，因此文献[12]将这种一阶相似性进一步拓展为 LINE 模型. 同时，将 DeepWalk 模型拓展到节点的二阶相似性学习，即假设节点 A，B 之间存在共享的相似节点，则节点 A，B 之间的相似性也就越高. 实验表明，这种拓展是有效的，并且获得的网络表达也更好.

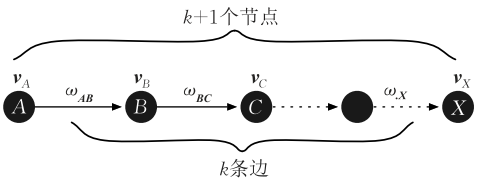


图 4　k 阶可达路径

在本文中，我们将网络的节点特征环境进行统一划分，将其定义为 k 阶可达路径. 如图 4 所示，从节点 A 到节点 X 存在一条有向带权路径，记

$$\begin{cases}P_{\boldsymbol{AX}}^{V}=\{\boldsymbol{v}_A,\boldsymbol{v}_B,\boldsymbol{v}_C,\cdots,\boldsymbol{v}_X\}\\ P_{\boldsymbol{AX}}^{\omega}=\{\omega_{\boldsymbol{AB}},\omega_{\boldsymbol{BC}},\omega_{\boldsymbol{C.}},\cdots,\omega_{\boldsymbol{.X}}\}\end{cases}\tag{3}$$

式(3)中，$P_{\boldsymbol{AX}}^{V}$，$P_{\boldsymbol{AX}}^{\omega}$分别为可达路径中的节点向量集合和权重集合. 此处权重集合由网络自身属性决定：对于带权网络，ω 值由输入源决定；对于非带权网络，$\omega=1.0$ 为恒定值.

值得注意的是，此处的 k 阶可达路径可以看做是上述 3 种特征环境的综合，每种特征的多少取决于不同的超参数 ε(4.1 节讨论)决定. 关于超参数 ε 对于网络特征学习的影响，将在实验部分的第 5.4.3 节(1)部分中进行分析.

3.2　动态链接预测问题

链接预测问题是网络分析中的基础问题之一. 对于社会网络而言，由于信息的不完整性以及网络的动态性，通常存在两种形式的链接预测问题，即当前时刻的未知链接预测(Missing Link Prediction，MLP)以及未来时刻的链接预测(Future Link Prediction，FLP)[19]. 一般的，有关链接的动态性考察，指的则是对于 FLP 问题的分析. 在现有的关于 FLP 问题的研究多数仅限制于链接的新增性的预测问题，即考察是否存在在未来的某个时刻出现某些链接. 因此，在本文的关于动态链接预测的考察中，我们同样仅考虑对 DNPS 模型以及基准模型在 FLP 的新增性问题上的准确性. 对于 FLP 问题及其评估定义如下.

假设存在网络 $G_t=(V_t,E_t)$，$G_{t+1}=(V_{t+1},E_{t+1})$分别对应于网络 G 在 t 和 $t+1$ 时刻的节点与边集合. 此时，考虑新增链接问题存在如图 5 所示 3 种情况.

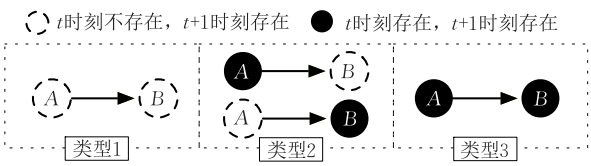


图 5　新增链接类型

考虑到对未来节点的不可预测性，现有的 FLP 问题通常限定于第 3 种类型的链接预测，即节点 A 与节点 B 在 t 时刻已经存在，但是它们之间的链接在 t 时刻以后才形成. 对于这种情形，预测模型通常通过计算已存在节点之间的相似性，来对未来可能形成链接的概率进行预测，即认为：如果两个节点在 t 时刻相似度较高，并且它们之间不存在链接，那么它们在后续时间内产生链接的可能性将会较大，这也是 FLP 预测问题的基本假设. 因此本文对于 FLP 问题考察如下指标：

AUC 准确性. *AUC* 准确性被广泛用于链接预测问题中[20-21]. 其计算方式如式(4)所示.

$$AUC=\frac{\sum_{i=1}^{n}S_i}{n} \tag{4}$$

其中：n 为测试次数；S_i 为单次测试得分. 记一次测试过程如下.

(1) 随机从 t 时刻，选取图 G 中不存在的边 e_t^{Train}，从 $t+1$ 时刻选取新增加的边 e_{t+1}^{Test}.

(2) 计算两条边通过模型 X 在 t 时刻给出的预测概率：

$$\begin{cases} p(e_t^{\text{Train}})<p(e_{t+1}^{\text{Test}})\rightarrow S_i=1 \\ p(e_t^{\text{Train}})=p(e_{t+1}^{\text{Test}})\rightarrow S_i=0.5 \\ p(e_t^{\text{Train}})>p(e_{t+1}^{\text{Test}})\rightarrow S_i=0 \end{cases} \tag{5}$$

N-Rank 排序. *N-Rank* 指标主要考察模型对于特定预测比例链接的召回率. 相比于 *AUC* 指标而言，*N-Rank* 指标只能够作为模型的辅助参照. 和 *AUC* 计算方式相似的是过程(1)，在过程(2)时，*N-Rank* 通过对步 1 选取的 n 条边的评估概率进行排序，计算得到前 n 条边中 e_{t+1}^{Test} 存在的数目的比例，即

$$N\text{-}Rank=\frac{\text{Count}(e_{t+1}^{\text{Test}})}{n} \tag{6}$$

模型 *AUC* 值与 *N-Rank* 值的高低直接反映出模型在 t 时刻对于网络结构特征抽取的好坏，一般的 *AUC* 与 *N-Rank* 越接近于 1 表示模型越优秀. 本文将以 *AUC* 值为主要指标衡量模型的结构特征提取能力，*N-Rank* 值作为辅助参考.

4 基于阻尼采样的动态学习模型

本节将对 DNPS 模型进行详细的论述. 4.1 节与 4.3 节给出了基于阻尼的正负采样算法. 0 小节则在 4.1 节的基础上给出了基于新增边的局部搜索动态扫描算法. 4.4 节给出了基于阻尼采样的分布式学习模型的具体推导与优化过程. 4.5 节则对 DNPS 模型给出了一个整体性的算法流程描述与总结，同时从理论上对 DNPS 模型进行了复杂度分析.

4.1 基于阻尼的正采样算法

社会网络是具备层次性的[22]，这种层次性使得处在不同关系层次的节点对于同一个节点的影响程度不同[23]. 因此，本文在 DNPS 模型的学习过程中引入基于阻尼的衰减学习方法. 其基本思想是：在预测的过程中，对每一次预测给予一个动态的影响阻尼，这种影响越大，则表现在节点之间的关系越紧密，反之，它们之间的相互影响则更弱. 通过引入基于阻尼的学习策略，能够有效地把定义 3 中的 3 种不同节点特征环境进行统一，而不需要独立地针对每一种节点特征环境进行学习. 下面，对基于阻尼的正采样算法进行具体分析.

假设对于网络 G 中存在一条随机游走的路径 $A\rightarrow B\rightarrow C\rightarrow D\rightarrow E\rightarrow F$，考虑一次预测过程中由节点 A 到节点 D 的预测阻尼 δ_{AD}，其计算方法如式(7)所示.

$$\delta_{AD}=\delta_{AB}\cdot\delta_{BC}\cdot\delta_{CD}=\prod_{i\in Path(A\rightarrow D)}\delta_{\boldsymbol{X}_i\boldsymbol{Y}_{i+1}} \tag{7}$$

式(7)中，$\delta_{\boldsymbol{X}_i\boldsymbol{Y}_{i+1}}$ 为起始节点到终止节点之间的路径的阻尼值. 对于带权网络，设定全局默认阻尼为 ε，通常 $\varepsilon\in(0,1.0)$，则式(7)可以改写为式(8)所示.

$$\begin{aligned}\delta_{AD}&=(\omega_{AB}\cdot\varepsilon)\cdot(\omega_{BC}\cdot\varepsilon)\cdot(\omega_{CD}\cdot\varepsilon)\\&=\omega_{AB}\cdot\omega_{BC}\cdot\omega_{BC}\cdot\varepsilon^3\end{aligned} \tag{8}$$

式(8)中：ω 为网络的自定义权重，通常由网络的自身属性决定，对于非带权网络；$\omega=1.0$ 为恒定值. 即对于带权网络，式(8)可以缩写为 $\delta_{AD}=\varepsilon^3$，此时动态阻尼系数即可以理解为超参数 ε 的指数函数. 根据本文的定义，$\varepsilon\in(0,1)$，因此，对于固定可达路径距离的预测，对于不同的超参数 ε，其动态阻尼的衰减函数的变化幅度存在较大差异.

算法 1[①]. Damping-based Positive Sampling，DPS.

输入：网络 $G=(V,E)$，节点最低增量率 λ，最长覆盖路径 $L_{\max}$，最短覆盖路径 $L_{\min}$

输出：正采样训练对集合 $C_{\text{DPS}}=[(X,Y,\delta_{\boldsymbol{XY}})]$

```
1.  cover_count=set()
2.  def return_DPS(path): //返回正样例
```

① 本文中所设计的所有算法语法以 Python 2.7 作为参考.

```
for i in range(len(path)-1):
  for j in range(i+1, len(path)):
    δ_ij = pow(ε, j-i)×∏_i^j ω_i //计算动态阻尼
    yield(path[i], path[j], δ_ij)
while True:
 for seed in V: //遍历一次节点集合
  path=[seed] //初始化可达路径
  L_path=1
  while True:
   try:
    next_node=random.choice(seed.children)
   except IndexError:
    if L_path<L_min:
     break
    else: //调用函数,获取动态阻尼正样例
     C_DPS+=return_DPS(path)
     cover_count|=set(path)
     break
   path.append(next_node)
   seed=next_node
   L_path+=1    //计算路径长度
   if L_path==L_max:
    C_DPS+=return_DPS(path)
    cover_count|=set(path)
    break
 if(len(V)-len(cover_count))/len(V)<λ:
  break //计算覆盖率
```

以跳数(hops)=3 为例，当默认全局超参数阻尼 ε 值越大时，其动态阻尼值越大，同时整体衰减幅度越平缓，那么在这种情况下，也就越容易获得节点的深度特征(因为衰减平缓，所以长距离的特征才能得以保留). 相反的，如果超参数 ε 定义越小，当距离较长时，预测节点产生的贡献几乎可以忽略，例如，当 ε=0.1 时，跳数大于 3 的贡献仅为误差的 0.001，此时，宽度特征得以保留，而深度特征几乎被忽略. 所以，对于不同的网络，设定(或称调出)与网络本身结构特征相适应的 ε 值(使得宽度和深度特征得到一个合适的配比)，能够有效地在保持准确度的同时，降低训练时间.

环境共现节点的动态阻尼值计算是基于阻尼的正采样(Damping-based Positive Sampling，DPS)算法的核心步骤之一. DPS 算法主要用于产生训练所需的正例，通过对网络中节点路径进行覆盖和遍历，产生所需要的训练样例. 同时，需要注意的是，DPS 算法中的超参数节点最低增长率 λ 是影响训练速度和准确度的指标之一. 当 λ 的预定义值越小，训练时节点的覆盖路径也就越多，模型得到的训练也就更加充分，反之 λ 越大，则越容易提前终止训练样例的生成，从而导致训练结果并不理想. 但是，伴随着效果提升的同时，较小的 λ 也将带来较大的时间开销. 因此，在训练过程中需要根据模型的实际要求灵活选取，关于 λ 参数的讨论将在实验讨论部分(第 5.4.3 节(3)部分)进行详细分析.

4.2　基于局部搜索的增量学习正采样算法

结构的动态性变化是社会网络的重要特征之一，尤其对于大规模网络而言，网络的动态变化使得很多基于静态的网络特征学习和表示模型在学习效率上并不理想. 针对社会网络的动态性，本文提出了基于局部搜索的增量正采样策略. 通过对新增边进行分类后，采用特定增量采样算法和初始化策略，快速重构得到新增节点的特征表达.

如图 6 所示，对于示例网络在 t 时刻的链接状态为实心节点及其实线，$t+1$ 时刻的新增状态为空心节点及其虚线. 单从新增边的角度考虑，存在 3 种形式，3.2 节已概述. 针对每种新增边的类型，DNPS 模型设计了对应的新增边采样算法，如算法 2 所示. 基于局部搜索的动态正采样算法的目的是在最小的时间开销内，取得必须的结构变更信息. 例如，对于类型 1，独立的新增边需要通过对新增边附近所涉及的其它新增节点的关系进行重新的随机预测来获取有新增边带来的结构改变信息. 而对于类型 2 中的新增节点而言，则需要以新增节点为中心，反复覆盖新增节点周边的新增可能路径.

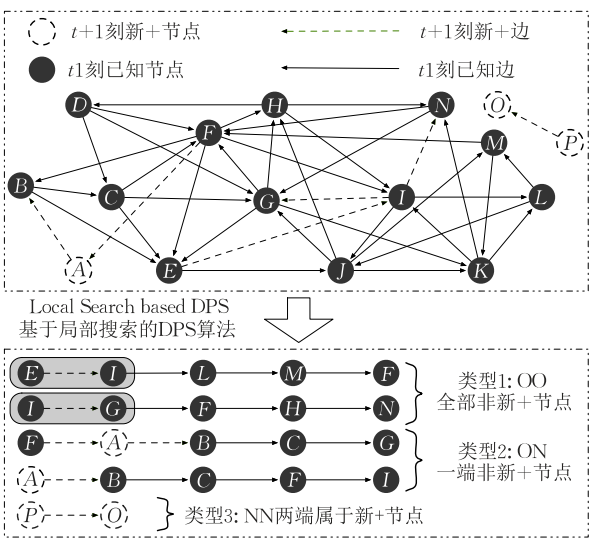


图 6　局部搜索的 DPS 算法示例

算法 2. Local Search based DPS, LSDPS.

输入：新增边列表 E_{new}，新增节点 set V_{new}

输出：新增正采样训练对集合 $C_{DPS}=[(X,Y,\delta_{XY})]$

```
def get_path(from_node, to_node):
  path=[from_node, to_node] //初始化增量路径
  L_path=2
  while True:
   try:
    next_node=random.choice(to_node.children)
   except IndexError:
    return(path)
   path.append(next_node)//随机选择路径方向
   to_node=next_node
   L_path+=1
   if L_path==L_max: //判断是否达到预置最长路径
    return(path)
def return_LSDPS(path, type):
  if type==0://计算新增节点类型 1(OO)的正采样
   for i in range(len(path)-1):
    for j in range(i+1, len(path)):
     δ_ij=pow(ε, j-i)×∏_i^j ω_i
     yield(path[i], path[j], δ_ij)
  elif type==1://计算新增节点类型 2(ON)的正采样
   if path[0] in V_new:
    for i in range(1,len(path)):
     δ_0i=pow(ε,i) * ∏_0^i ω_0i
     yield(path[0], path[i], δ_0i)
   else:
    yield (path[0], path[1], ε·ω_01)
    for i in range(2,len(path)):
     δ_ij=pow(ε, i-1)×∏_1^i ω_1i
     yield (path[1], path[i], δ_1i)
  else: //计算新增节点类型 3(NN)的正采样
   yield(path[0], path[1], ε·ω_01)
for link in E_new: //对新增边进行扫描，判断边类型
  from_node, to_node=link.split()
  link_type=len ({from_node, to_node}&V_new)
  path=get_path(from_node, to_node)
  C_DPS+=return_LSDPS(from_node,to_node,link_type)
```

算法 2 是在算法 1 的基础上，针对新增节点动态结构的出现所进行的增量采样. 实验表明，算法 2 所获取的动态变更信息能够有效地在保证较少时间开销的同时，维持甚至提升算法的 *AUC* 准确率，这种高效性能够使得模型在大规模动态的社会网络环境下能够有效地适应网络变化特征.

4.3 网络节点特征学习的负例选取

通过获取得到节点的阻尼正采样后，在此基础之上，构建其对应的负采样节点列表，从而对网络节点特征的分布式表达进行学习. 对比于正采样算法而言，负采样算法相对简洁. 其基本问题是：对于已给定的节点 A 而言，如何产生训练所需的负采样节点集合 N_A. 此处，对于每一个正样例所匹配的负样例数目 $|N_A|$ 为本文的超参数之一，需要根据网络的结构特征进行调整. 关于超参数 $|N_A|$ 对于 DNPS 模型训练时间以及效果的影响将在第 5.4.3 节(4)部分中进行分析和讨论. 序列的负采样[24]学习，较早由 Mikolov 等人[7]基于词向量的训练而提出，用以加速词向量的训练速度，并同时提升向量的训练效果. 我们同样采用类似的负采样方法应用于社会网络节点向量的训练过程中. 在原有的负采样方法中，作者将其定义为一个带权采样问题，即通过统计已出现的词语的词频，对于词频较大的词语，在负采样时，使得被获取的几率应该较大. 因此定义如下的采样函数：

$$p(w_i)=\frac{Freq(w_i)}{\sum_{j\in D}Freq(w_j)} \tag{9}$$

式(9)中，$p(w_i)$ 可以理解为带权归一化的概率值.

通过式(9)的遍历计算后，即可获得每个词语在一次随机的负采样中，被选中的归一化概率. 然而，我们在实际的研究中发现，虽然这种负采样方法在文本向量的学习过程中效果较好，但是在社会网络节点的学习过程中却并不理想. 其主要原因是，相对于词语而言，社会网络中更容易存在“富者越富”的聚集现象.

例如，当针对节点 A 进行负采样学习时，如果单纯地直接采用式(9)进行负例的选取，此时节点的度越大则越容易被选到，而此时往往节点 A 与大度节点形成连接的可能性更大，那么此时的负例的选取就存在很大可能性的失败. 即，相对于文本网络而言，社会网络的不平衡性更加明显. 因此，我们在构建上述负采样表时，采用了对应的修剪策略，即直接剔除节点频率排名为前 20% 的节点，而选用剩余后 80% 的节点作为负采样节点的备选列表. 通过我们的初步实验表明，这种采用了修剪模式的负采样策略，在社会网络中比直接的负采样策略效果更好. 同时，此处 20% 的删除比例为本文模型的预设参数，一般的，该参数可以根据网络的实际分布情况在

10%～40%间调整. 删除比例越小，相对时间开销越大，反之，时间开销将会有所降低.

4.4 基于正负采样的学习模型推导

本文中，基于正负采样的网络节点连续特征表达学习可以看成一种动态逻辑回归模型，通过学习节点之间的共现关系，使得节点之间的结构信息能够有效地被蕴涵到实数向量空间. 基本思路为在已知节点 A 的正采样序列 P_A 的前提下，考虑如何有效地利用正采样序列中的节点去预测节点 A 本身存在(可达)的概率. 同时，每一次选取正采样训例时，都匹配以动态的负采样训例. 然后，采用随机梯度算法，得到的误差，反向修改初始化的向量空间，最后得到节点的有效向量表达. 下面将结合一个例子，对 DNPS 模型的目标函数以及优化更新方法进行推导.

假设在一次随机正采样中，选取了节点 A 及其对应的正采样序列 P_A. 那么，对于节点 A 及其 P_A 中的节点，存在式(10)所示的条件预测概率.

$$p(A|B) \tag{10}$$

此时，通过查询可以得到节点 A,B 的向量表达为 $\boldsymbol{v}_A$ 和 $\boldsymbol{v}_B$($\boldsymbol{v}_A,\boldsymbol{v}_B\in\mathbb{R}^d$)，由此式(10)的概率计算可以转变为

$$p(\boldsymbol{v}_A|\boldsymbol{v}_B) \tag{11}$$

对于式(10)而言，我们的目标是求得使其最大化的参数(向量)集合，即式(11)的最大化.

类似的，此时相对于正采样集合 P_A 而言，我们可以构造得到其负采样集合，记为 N_A，对于此时当从 N_A 中取一个负采样节点 K，同样可以定义由 K 预测得到节点 A 的概率为式(12).

$$p(A|K) \tag{12}$$

对于式(12)而言，与式(11)相反的是：我们的目标是求得使其最小化的参数(向量)集合. 联合式(10)和(12)，给定 P_A 以及 N_A，对于节点 A 而言，目标函数可以改写为式(13)所示.

$$o(A)=\prod_{j\in P_A}\prod_{i\in\{A\cup N_j\}}p(i|j) \tag{13}$$

式(13)中，$p(\cdot|\cdot)$ 为分类概率函数，其计算方法如式(14)所示.

$$p(i|j)=\begin{cases}\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i), & j\in P_i\\ 1-\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i), & j\in N_i\end{cases} \tag{14}$$

式(14)中，$\sigma(\cdot)$ 为逻辑回归函数，即 $\sigma(x)=\dfrac{1}{1-e^{-x}}$，$\boldsymbol{\theta}^i$ 为辅助向量，与节点向量 $\boldsymbol{v}$ 的维度数相同.

结合式(14)可以看到，对于目标函数(13)的含义为：在给定节点 A 时，最大化其环境节点内部预测得到的概率，同时最小化其负采样节点预测得到的概率的问题可以统一为式(13)的最大化. 联合式(13)和(14)可得式(15).

$$\begin{aligned}o(A)&=\prod_{j\in P_A}\prod_{i\in\{A\cup N_j\}}p(i|j)\\&=\prod_{j\in P_A}\prod_{i\in\{A\cup N_j\}}[\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i)]^{\alpha}\cdot[1-\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i)]^{1-\alpha}\end{aligned} \tag{15}$$

式(15)中，α 满足 $j\in P_i$ 时 $\alpha=1$；$j\in N_i$ 时 $\alpha=0$.

可以看到，式(15)为当选定某特定节点时关于该节点的目标函数，那么对于网络中的所有节点集合 V 而言，其总体目标函数可以转化为式(16)所示.

$$O(V)=\prod_{A\in V}\prod_{j\in P_A}\prod_{i\in\{A\cup N_j\}}p(i|j) \tag{16}$$

将式(15)代入式(16)，同时，对目标函数取对数可得式(17).

$$\begin{aligned}OL=\log(O(V))&=\log\Big(\prod_{A\in V}\prod_{j\in P_A}\prod_{i\in\{A\cup N_j\}}p(i|j)\Big)\\&=\sum_{A\in V}\sum_{j\in P_A}\sum_{i\in\{A\cup N_j\}}\log p(i|j)\\&=\sum_{A\in V}\sum_{j\in P_A}\sum_{i\in\{A\cup N_j\}}\log([\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i)]^{\alpha}\cdot\\&\quad[1-\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i)]^{1-\alpha})\end{aligned} \tag{17}$$

为简化推导过程，将式(17)简记为式(18)所示.

$$\begin{aligned}OL(A,j,i)&=\log([\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i)]^{\alpha}\cdot[1-\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i)]^{1-\alpha})\\&=\alpha\cdot\log[\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i)]+(1-\alpha)\cdot\\&\quad\log[1-\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i)]\end{aligned} \tag{18}$$

此时目标函数的最优化(最大化)可以转换为式(18)的求偏导问题. DNPS 模型的对于式(18)的优化问题同样采用较为快速的随机梯度上升的方法.

由式(18)分别对 $\boldsymbol{\theta}^i$ 以及 $\boldsymbol{v}_j$ 求偏导数，可以得到两者的更新梯度分别式(19)和式(20)所示. 此处优化梯度的推导方式和文献[13]中的推导方法相似，故此处简略给出推导结果.

$$\boldsymbol{\theta}^i:=\boldsymbol{\theta}^i+\eta[\lambda-\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i)]\boldsymbol{v}_j \tag{19}$$

$$\boldsymbol{v}_j:=\boldsymbol{v}_j+\eta\sum_{j\in\{A\cup N_j\}}[\lambda-\sigma(\boldsymbol{v}_j^{\mathrm{T}}\boldsymbol{\theta}^i)]\boldsymbol{\theta}^i \tag{20}$$

式(19)及(20)中，η 为学习速率.

4.5 模型总结

4.1 节至 4.4 节对模型的主体框架进行了独立阐述，本小节将在其基础上进行总结与分析，并给出算法的总体训练流程与模型的复杂度分析.

4.5.1 模型训练

DNPS 模型首先根据网络的初始状态，通过正负阻尼采样算法产生对应的训例. 然后根据训例对

随机初始化的网络节点向量结合随机梯度上升进行修改，通过寻求目标函数的优化值，得到优化的向量参数组合. 同时，针对动态社会网络的特征，DNPS基于局部搜索的正采样算法构建增量训例，通过对不同的新增节点类型进行学习，以较短的时间获取较好的增量式表达. 模型训练的流程化表达如算法 3 所示.

算法 3. DNPS 模型训练框架.

输入：网络边列表(初始边列表或增量边列表)

输出：节点的特征向量

```
if not Incremental_Learning: //非增量学习
  Network Preprocessing //网络预处理
  Parameters initialization //参数初始化
  DPS 算法 //基于阻尼衰减的正采样算法
  for (X,Y,δ_XY) in C_DPS
    Eg=0
    for i in N_Y ∪ X:
      if i==X: λ=1
      else: λ=0
      Eg+=δ_XY×η[λ−σ(v_Y^T θ^i)]θ^i
      θ^i+=δ_XY×η[λ−σ(v_Y^T θ^i)]v_Y
    v_Y+=Eg
else: //增量学习
  Edge Classification //新增边分类
  LSDPS 算法 //基于局部搜索的阻尼衰减
  for (X,Y,δ_XY) in C_LSDPS:
    Update Parameter Vectors//与非增量相同
```

从算法 3 中可以看到，模型的训练时间在一定程度上和正负采样的样例数相关. 通过对所有的正负样例进行随机遍历，以此优化参数向量组合. 模型的迭代过程包含在样例的选取过程中. 同时，在增量学习的实际编程中，我们提供了包含本文描述在内的快速增量学习(LSDPS)以及基于迭代的增量学习两种训练模式，供其他研究者在效率和准确度之间灵活选取.

4.5.2 模型复杂度分析

(1) 空间复杂度分析

DNPS 模型对于网络特征的存储采用节点连续向量模式，对于一个节点向量为 d 维的网络，其核心存储空间开销为 $|V|\cdot d$. 特别是对于大规模网络，通常有 $|V|\geq d$，因此 DNPS 模型对于大规模社会网络而言具有较小的内存开销. 换句话说，一旦模型训练完成，只需要存储容量为 $|V|\cdot d$ 的节点特征网络即可用作其它应用的基础.

当然，在训练过程中，由于模型还涉及到辅助向量参数 θ 的存储以及其它内存开销. 同时，如果需要保留全部模型参数，以用于后续的增量训练，通常模型的空间开销如式(21)所示.

$$S(n)=O(dn) \tag{21}$$

其中：d 为向量的维度；n 为网络中的节点数. 相比于现有的较多关于矩阵分解的模型而言，DNPS 模型能够有效地降低内存的开销，使得大型网络的分析和研究在较小内存环境下可行.

(2) 时间复杂的分析

考虑到采样期间与训练期间的时间开销，如式(22)所示.

$$T(n)=d\times(|N_X|+n\times\lambda) \tag{22}$$

其中：$|N_X|$ 为平均负采样个数；λ 为最小采样覆盖率. 值得注意的是 λ 的大小将会影响正采样的次数，同时也会影响对应的负采样总次数. 一般的，λ 越小，模型的训练效果越理想，节点的结构特征的提取也就越充分，但也会带来时间的增加. 具体的影响方式，将在重要超参数分析一节给出.

5 实验设置与分析

5.1 实验设置

本文所有模型(包含对比模型)的统一实验环境的主要设置如表 3 所示. 实验过程中所有模型的共有超参数遵循统一控制变量的原则，通过设置基准参数，对不同模型的准确率以及时间效率在相同的框架内进行对比分析. 在实验分析部分将进一步对模型(或对比模型)的效果以及其超参数的影响进行详细阐述.

表 3 实验系统设置信息

项目	设置	数量
操作系统	Yosemite 10.10.5	1
CPU	Intel i7-5280k，6 核，12 线程	1
硬盘	512GB PLEXTOR PX-512M6Pro SSD	1
内存	Kingston 8GB DDR4 2400	8
重要程序包	Python 2.7.11，Cython 0.23.4	1

5.2 数据集分析

5.2.1 数据集信息

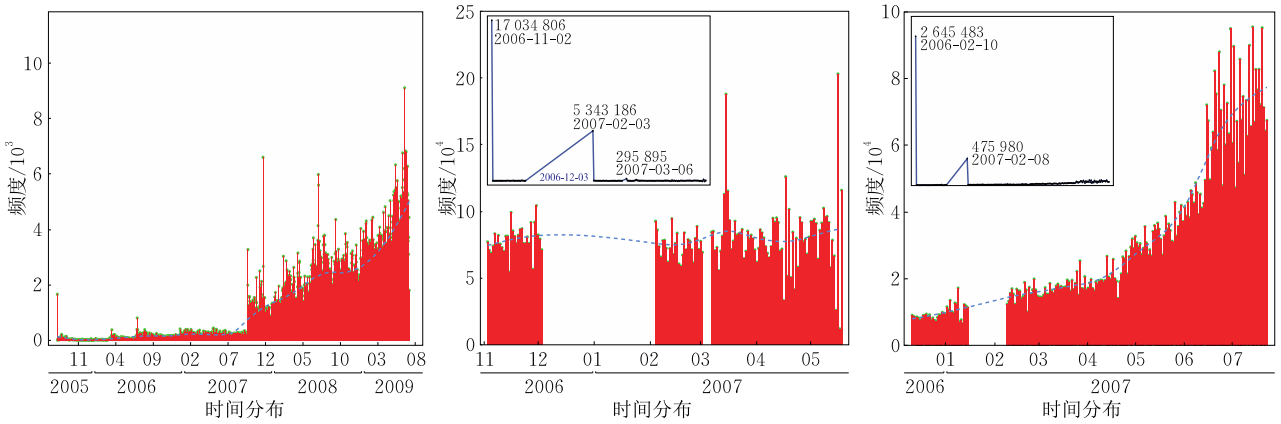
考虑到针对社会网络的动态变化过程进行分析，需要选取含有时间戳的社会网络数据集. 因此，我们选取了当前被广泛采用的 3 个大规模动态网络数据集：Digg 网络、Flickr 网络和 YouTube 网络. 其拓扑结构的基本统计信息如表 4 所示. 同时，将上述 3 个网络以“天”为单位，计算每天新增边的数目后得

到了如图 7 所示的网络新增边频度分布. 其中，由于 Flickr 网络以及 YouTube 网络的原始边数相比于新增边数占有绝对数量优势，为了更好地显示新增边数的分布情况，我们将起始边数以及部分缺失数据导致的节点数大幅增长情况体现在左上角的折线图中.

表 4　数据集基本信息

数据集	方向	节点数	边数	平均度	MSPL
Digg	D	279 630	1 731 653	12.3850	4.31
Flickr	D	2 302 925	33 140 017	28.7810	5.46
YouTube	UD	3 223 589	9 375 374	5.8167	5.29

注：表中 MSPL 为平均最短路径.



(a) Digg　(b) Flickr　(c) YouTube

图 7　数据集新增边频度分布图

Digg 网络. Digg 网站[①]为著名的新闻类“掘客”网站，网络中的每一个用户都可以成为该网站中的一名“掘客”. 当有新的文章(新闻)产生时，用户可以对该文章进行投票，标注以及评论(Digg 一下). 当文章的 Digg 数目达到一定的级别上，该文章就会被推荐到网站的首页或者其他用户的主页上. 本文实验的 Digg 网络[②]从 2009 年开始收集，总共包含了约 27 万用户之间的有向关注关系.

Flickr 网络. Flickr 网站[③]是以图片寄存，标记以及搜索为主题的著名图片分享类社交网站. 其主要特点为以社会网络为核心来进行社会关系的拓展与内容的分享与组织. 用户可以在自身账户中加入联系人(Friends)，并且可以针对不同的关系对照片内容进行组织和分类. 本文实验的 Flickr 网络[④]包含了从 2006 年 11 月到 2007 年 5 月期间约 230 万用户的有向朋友关系.

YouTube 网络. YouTube 网站[⑤]为目前世界上最大的视频分享类网站. 用户可以往 YouTube 上传和分享自己或朋友的视频，同时可以关注他人的视频更新或者分享状态，以此形成朋友关系. 本文实验的 YouTube 网络[⑥]为 2006 年 12 月到 2007 年 7 月期间形成的约 322 万用户之间的无向朋友关系.

5.2.2　实验评测数据划分

在对数据集进行清洗和统计的过程中我们发现：原始采集者在数据的采集过程中存在间断，即并不是完整的采样. 例如 YouTube 数据集采集的是从 2006-12-10 开始至 2007-07-23 结束时的网络动态变化，其中每次的采集间隔为 1 天. 但是在 2007-01-16 至 2007-02-07 之间的动态变化却并没有记录下来. 如图 7 所示，Flickr 和 YouTube 数据集均存在“断档”的情况. 因此，我们对于上述数据集的动态链接预测问题采用按等长日期为划分进行验证. 划分后的统计数据信息如表 5 所示.

表 5　迭代验证数据集划分

网络	Digg		Flickr		YouTube	
	V^{AP}	E^{AP}	V^{AP}	E^{AP}	V^{AP}	E^{AP}
$T0$	1278	1674	1 834 425	24 828 240	1 406 185	3 465 249
$T1$	4718	7677	44 361	843 134	119 418	278 569
$T2$	7364	10 609	40 831	790 926	125 444	293 516
$T3$	12 004	24 132	54 661	1 051 474	138 954	333 830
$T4$	14 143	30 058	103 997	1 067 729	134 663	353 800
$T5$	16 379	34 361	47 553	836 825	180 906	513 664
$T6$	32 732	129 715	43 791	905 219	201 969	605 454
$T7$	40 892	225 762	38 115	768 037	224 453	762 683
$T8$	42 401	342 681	42 758	926 192	333 713	1 254 053
$T9$	39 418	386 097	46 606	994 416	311 273	1 311 588
$T10$	68 301	538 887	5827	127 825	46 609	202 969

表 5 中，$T0$ 为网络的初始状态所包含的节点数以及边数，$T1 \sim T10$ 为网络动态变化时，当前网络状态相对于上一个时刻状态新增的边数以及节点

① http://digg.com
② http://konect.uni-koblenz.de/networks/digg-friends
③ http://flickr.com
④ http://konect.uni-koblenz.de/networks/flickr-growth
⑤ http://YouTube.com
⑥ http://konect.uni-koblenz.de/networks/YouTube-u-growth

数.考虑到基准数据集获取时所采用的方法并没有将节点消失以及边的消失纳入考虑范围,因此本文暂时只考虑边的新增预测问题.从表 5 可以看到,网络的新增边数要远大于新增节点数目.

5.2.3 实验评测过程

如图 8 所示,为本文动态网络链接预测的评测流程,其基础思想为以 T 时刻状态以前的网络为基准,对 $T+1$ 时刻的网络状态进行预测.此时考虑模型对于动态变化的适应性,可以分为两类模型:增量式计算以及非增量式计算.增量式计算则利用 $T-1$ 时刻及其以前的训练成果,通过动态融入 T 时刻的新增信息后对 $T+1$ 时刻的网络状态进行预测,而非增量式则需要在 T 时刻重新训练或计算整个模型.通常而言,非增量式模型时间开销更大.本文在实验评估时,将分别对上述两种模式进行考察.

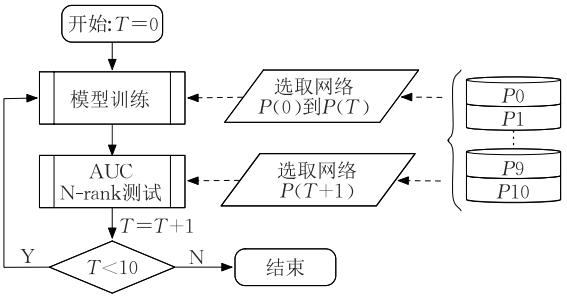


图 8 网络动态链接预测评测流程

5.3 基准对比方法

在实验的过程中,本文选取了目前取得效果较好的网络特征学习算法,分别从算法的运行效率以及预测的准确性两个方面,在动态社会网络的链接预测问题上进行评估.选取的核心对比算法如下.

DeepWalk.该算法在 2014 年由 Perozzi 等人[9]发表于 KDD 国际会议.DeepWalk 借鉴了深度学习在文本中应用 Word2Vec 的生成过程,通过在网络中进行较短的随机游走来获取网络的拓扑结构特征,以获得网络的特征表示向量.和传统的网络特征表示学习方式相比,该方法能够有效应对网络数据的稀疏性特征,并在数据量较少的情况下取得同等的优秀结果.

LINE.该算法在 2015 年由 Tang 等人[12]发表于 WWW 国际会议.LINE 算法在 DeepWalk 的基础上,加入了对节点之间的二阶关系的分析.从更深层次的角度获取节点之间的拓扑结构特征关系.同时,LINE 针对一阶和二阶关系设计了与之对应的目标函数进行优化,然后对得到的两种类型的向量进行拼接组合操作,由此获得最后节点特征向量的表达.LINE 模型包含了 3 个子模型,分别是 LINE-1st、LINE-2nd 以及 LINE(1st+2nd)模型.在对比实验中,我们将同时考察上述 3 个子模型.

本文将选取上述两个主要模型在预定的优化参数设置上进行分析,并且给出上述基准模型以及 DNPS 模型在实验数据集上的动态链接预测结果.

5.4 实验结果与讨论

为了有效地对比 DNPS、DeepWalk 和 LINE 模型在社会网络结构特征学习后用以动态链接预测问题上的效果,我们将分为两部分进行评估.首先是基于单一时间窗口评估,用以在时间点上对模型的特征提取能力进行分析对比.其次是基于连续时间窗口的评估,用以考察模型在大规模动态社会网络上的适应能力,即在连续变化的问题上,进行分析.在上述两种分析过程中,我们都将同时给出模型预测的 AUC 准确率以及时间开销的具体得分.最后,我们将对 DNPS 模型中涉及到的主要核心超参数进行影响分析,给出实践过程中 DNPS 模型核心超参选取的建议,帮助读者在实际应用中快速选取适合动态网络的较优超参数组合.

5.4.1 单一时间窗口

由于 LINE 模型以及 DeepWalk 模型在设计过程中,并未给出独立的增量式训练方法.因此,为了有效对比模型,下面首先在静态的单一时间窗口设定下进行分析.如表 6 所示,为单一时间窗口下 AUC 准确率及其对应的时间开销结果.需要注意的是 LINE 模型以及 DeepWalk 模型同样存在超参数的设置问题,我们根据作者论文提供的描述,进行了相应的超参数调试实验,选取了在可接受时间范围的最佳实验结果.例如,对于 LINE 模型,如作者在文中所提出的,通过增加 Samples 的数量可以提升模型的预测效果,但是在实际的实验过程中发现,Samples 的增加会导致实验时间的大规模提升.因此,对于 LINE 模型我们在阐述其最佳 AUC 值的同时给出相应的

表 6 单一时间窗口准确率 *AUC*/%与时间/s 开销

数据集	结果	LINE-1st	LINE-2nd	DeepWalk	DNPS
Digg	*AUC*	36.139	73.792	71.061	79.615
	N-Rank	19.7	26.6	15.58	29.4
	Time	227.21	254.07	21.3	15.6
Flickr	*AUC*	78.377	59.504	86.788	89.989
	N-Rank	15.1	8.7	7.79	11.3
	Time	343.8	4943.1	5336.5	592.3
YouTube	*AUC*	83.32	54.47	80.620	89.081
	N-Rank	34.9	48.7	24.27	23.3
	Time	412.4	3156.4	5845.8	353.2

时间开销进行对比. 通常认为，对于相近时间开销的模型，*AUC* 值越高，模型越好. 而当 *AUC* 相近时，时间开销越短，模型越好.

如表 6 所示，为 3 个大规模网络中的单一时间窗口下未来链接预测结果. 其中，黑体加粗包含下划线的数据为本行数据中的最大(*AUC*/*N-Rank*)以及时间开销(*Time*)最小的值. 基于表 6 可以发现：(1) LINE-2nd 模型在 Digg 网络以及 Flickr 网络中都取得了次优的结果，但同时由于 LINE-2nd 模型的复杂性较高，其计算的时间开销相比 DeepWalk 以及 DNPS 模型大. 例如，在保证训练精度的条件下，相对于 Digg、Flickr 以及 YouTube 网络，DNPS 对比 LINE-2nd 模型的训练速度分别有约 33 倍、8 倍和 9 倍的提升；(2) DeepWalk 模型在数据集较小时，训练速度较为理想，但是相对于 LINE 模型以及 DNPS 模型，其训练效果却并不好. 特别是对于 *N-Rank* 指标，DeepWalk 在 3 个数据集上的得分都为最低得分；(3) DNPS 模型在相同的时间开销的情况下，取得的 *AUC* 值较 LINE 模型以及 DeepWalk 模型都有较大的提升. 特别是对于大规模的网络，无论是 Flickr(有向)还是 YouTube(无向)，DNPS 都取得了较好的预测结果，同时 DNPS 还能够以较低的时间开销在大规模的网络中进行训练.

5.4.2　连续动态时间窗口

社会网络的动态性变化给结构特征的表达学习带来了较大的挑战，为了进一步分析 DNPS 模型以及对比模型在大规模社会网络动态性问题上的优劣，本节将按照实验设置对网络进行动态分段预测实验. 值得注意的是：DeepWalk 以及 LINE 模型都是基于非增量式学习的设计，无法利用作者提供的现有代码进行增量训练. 因此只能采用循环迭代的方式进行评测，即对每一次的新增网络采用重新学习的模式进行训练.

如表 7 所示，为连续动态时间窗口变更环境下的社会网络动态特征提取用以链接预测的实验结果. 其中，表的行标题 T0x-Ty，表示基于 T0 到 Tx 时间的网络数据集对 Ty 时刻新增的边进行的预测. 例如 T06-T7 表示使用 0～6 时间段内的网络数据集，对时刻 7 出现的边进行测试. 同时，上述所有模型都是基于 8 线程的并行计算，以此统一衡量模

表 7　连续动态时间窗口(*AUC*/%/*Time*/s)

0.000*：时间最短　0.000#：时间最长　0.000^：*AUC* 最高　0.000~：*AUC* 最低

Network	模型	指标	T01-T2	T03-T3	T03-T4	T04-T5	T05-T6	T06-T7	T07-T8	T08-T9	T09-T10
Digg	LINE-1st (1000M)	*AUC*	30.952	43.193	45.351~	45.115~	42.585~	39.816~	37.410~	36.139~	35.494~
		Time	174.31	193.06	178.57	190.35	202.20	215.72	239.64	227.21	251.19
	LINE-2nd (1000M+Re.)	*AUC*	55.826	52.235	59.401	59.064	59.754	62.818	65.634	73.792	76.322
		Time	169.60	168.76	200.68	198.11	211.00	224.56	242.69	254.07	267.68
	LINE (1st+2nd)	*AUC*	57.985	56.124	62.398	64.955	68.325	66.915	69.221	73.937	78.154
		Time	398.25#	416.04#	437.60#	441.52#	468.32#	491.24#	530.19#	533.49#	567.87#
	DeepWalk	*AUC*	21.652~	38.782~	50.021	58.416	62.100	65.982	68.102	71.061	72.834
		Time	0.81*	1.73*	2.81*	3.80*	6.16*	10.20*	16.03	21.27	26.06
	DNPS	*AUC*	68.614^	68.522^	68.401^	66.160^	70.559^	72.457^	72.830^	80.604^	83.709^
		Time	2.19	3.32	4.10	6.07	9.84	14.52	15.29*	15.30*	23.20*
Flickr	LINE-1st (1000M)	*AUC*	78.619	78.963	79.381	79.981	80.260	80.977	81.704	81.274	81.436
		Time	347.74*	349.56	377.85	355.49	358.68	377.89	369.72	381.67	383.24
	LINE-2nd (1000M+Re.)	*AUC*	59.824~	59.350~	59.310~	58.484~	58.421~	58.499~	59.255~	58.713~	59.025~
		Time	5437.02	5705.23	6075.34	6616.51	6918.96	7161.98	7424.20	7694.03	7921.55
	LINE (1st+2nd)	*AUC*	80.336	82.684	82.319	82.855	83.145	83.669	84.564	84.436	84.447
		Time	5785.24	6077.25#	6499.38#	6987.22#	7305.98#	7599.16#	7819.67#	8136.23#	9365.32#
	DeepWalk	*AUC*	87.309	86.617	86.446	86.790	87.080	87.132	86.932	87.317	87.232
		Time	5889.62#	5999.34	6245.45	6632.07	6877.17	7016.57	7328.86	7731.55	8066.50
	DNPS	*AUC*	89.876^	90.459^	91.638^	92.049^	92.329^	92.896^	92.947^	92.939^	93.049^
		Time	647.43	234.95*	228.71*	288.56*	288.32*	242.39*	253.46*	227.85*	264.43*
YouTube	LINE-1st (1000M)	*AUC*	83.769	83.110	82.695	82.781	83.364	82.954	82.317	82.854	82.365
		Time	334.50	369.25	365.57	374.32	385.14	372.21	395.25	414.86	425.98
	LINE-2nd (1000M+Re.)	*AUC*	53.426~	49.995~	48.913~	48.124~	47.658~	47.365~	46.897~	46.147~	45.104~
		Time	3616.91	4222.03	5040.92	5924.24	6754.26	7254.33	8024.60	8854.29	9014.79
	LINE (1st+2nd)	*AUC*	84.367	84.224	84.891	85.235	86.645	85.204	86.480	84.253	84.002
		Time	4031.25	4621.19	5460.05	6348.72	7186.58	7679.64#	8472.35#	9335.74#	9515.49#
	DeepWalk	*AUC*	81.020	81.231	81.537	81.685	81.725	80.993	81.294	81.448	80.836
		Time	5876.72#	6340.63#	6536.42#	6936.74#	7294.81#	7634.52	7738.18	8785.20	8925.77
	DNPS	*AUC*	86.338^	87.274^	87.461^	86.910^	86.735^	87.327^	87.400^	87.398^	88.058^
		Time	280.53*	96.98*	107.41*	120.93*	172.71*	204.31*	266.60*	319.48*	312.57*

型的时间开销. 其中,LINE 以及 DeepWalk 的向量维度均为 128.

首先,基于时间开销进行分析. 从表 7 中可以看到 LINE 模型在同样的并行(8 线程)环境下,所花费的训练时间最高. 其中,DeepWalk 在 Digg 网络占据了两个最高时间开销,其它最高时间开销全部由 LINE 占据,同时,还可以看到,DeepWalk 模型的时间开销同样非常大,与 LINE 模型在多个数据测试下非常接近. 特别是对于 YouTube,这类无向网络,其时间的开销相比于有向网络(例如 Flickr)而言更大,主要原因在于:从模型的设计以及代码的编写上来说,DeepWalk 以及 LINE 模型并未针无向网络进行优化,而只是简单认为"将无向网络的边更改为双向网络输入"即用以模型训练.

然而,对于小型网络而言,将无向网络双向化后作为模型的输入所导致的时间开销影响或许并不明显,但对于大规模网络,例如,百万、千万甚至上亿节点的无向网络,简单的双向化所带来的时间开销甚至将非常可观. 由表 7 可以看到,虽然 Flickr 网络的边数以及节点数要远多于 YouTube 网络,但是 DeepWalk 以及 LINE 模型在这两个数据集上的时间开销却非常相近. 其次,就 LINE 模型本身而言,LINE-2nd 子模型所花费的时间开销要远远地多于 LINE-1st 子模型. 通过对作者提供的源代码以及论文进行分析,我们发现,其主要原因为两点:首先,LINE-2nd 子模型从训练机制上比 LINE-1st 子模型更为复杂,其次就是 LINE-2nd 在训练过程中,为了有效地对低频次的节点进行处理,设计了针对稀疏网络的"Reconstructed 机制". 虽然该机制对模型的训练结果有一定的提升,却较大地影响了模型训练的时间开销.

最后,回归到解决社会网络结构变化的动态性问题本身而言. 由于缺乏有效的增量学习机制,无论是 DeepWalk 模型还是 LINE 模型,都只能针对新出现的网络状态变化进行重新学习. 从而使得它们的训练时间随着网络的动态增长而同步快速的增加. 不同的是,对于 DNPS 模型而言,由于设计了基于局部搜索的增量式学习算法,在应对网络的动态变化时,并不需要"从零开始",而是在当前的训练结果的基础上,针对新出现的节点或者边进行局部搜索学习,同时进一步优化原有的训练结果,使得模型在保证训练精度的同时有效地节约训练时间. 从表 7 中标"*"部分的数据可以看到,无论是针对有向网络还是无向网络,DNPS 模型的训练速度都是较快的,并且这种优势对于大规模网络(YouTube 网络以及 Flickr 网络)而言更加明显.

其次,基于 *AUC* 得分进行分析. 表 7 中标"^"号部分与"~"号部分的数据分别为当前预时间窗口下的最高与最低 *AUC* 准确率. 从表中整体上可以看到,DNPS 模型在所有连续变动窗口内都取得了最佳的 *AUC* 值.

针对 LINE 模型在不同网络中的表现可以发现,当网络的平均集聚度系数较大时,LINE-1st 子模型对于 LINE-2nd 子模型的表现更为理想,因此 LINE-2nd 并不能够较好的适应于稀疏的大规模网络,这与 LINE 模型文中所实证的结果是统一的. 例如,对于 YouTube 网络,随着时间窗户的推移,网络的稀疏性增加的同时,LINE-2nd 子模型的训练结果的预测精度反而降低了,相比之下 LINE-1st 子模型的训练预测结果在一定程度上保持了相应的稳定性,甚至有少许的上升. 针对 LINE 模型,作者还提出了通过将 LINE-1st 与 LINE-2nd 拼接构建联合模型的思想,即表中的 LINE(1st+2nd)子模型. 可以发现,结合 LINE(1st)与 LINE(2nd)的训练结果所构建的 LINE(1st+2nd)子模型在动态链接预测的 *AUC* 结果上都有一定的提升.

在 *AUC* 结果的表现中,DeepWalk 的表现相对中庸,并没有取得较好的结果,同时表现也不是十分差. 相比之下,LINE 及其子模型在对应的不同数据集上表现不一致,既有较好的表现,也存在表现很差的情况. 例如,LINE-2nd 在 Flickr 网络以及 YouTube 网络上的表现都不理想. 而 LINE(1st+2nd)表现则较为不错. 当然,其中 DeepWalk 在 Flickr 网络中也取得了较为不错的结果,但是由于训练的时效性较低,导致其对大规模社会网络的动态性无法有效的适用.

总的来说,LINE 模型以及 DeepWalk 模型在动态社会网络结构结构特征的表示学习过程中,时效性以及特征提取的稳定性是存在的核心问题. DNPS 模型针对上述问题设计了基于阻尼采样以及基于局部搜索的增量学习算法,有效的解决了大规模动态社会网络的结构特征的学习问题,在多个数据集上表现出了较高的时效性以及 *AUC* 准确率.

5.4.3 核心超参数分析

本小节主要对 DNPS 模型中所涉及的主要超参数对模型在准确率以及时间开销上所造成的影响进行分析. 在实验过程中,选用的默认超参设置为 $\varepsilon=0.9$,$d=128$,$\lambda=0.005$,$|N_x|=2$. 在对特定超参

进行实验时，其它参数保持默认不变，其中默认 *N-Rank* 排序时 $N=1000$.

(1) 阻尼系数 ε

如图 9 所示，为 DNPS 模型的 *AUC* 与 *N-Rank* 值随着阻尼系数的变更而发生改变示意图. 对于 Flickr 网络和 YouTube 网络而言，从图 9 中可以明显看到 *AUC* 值的变化趋势为典型的先上升后下降类型. 同时，Flickr 和 YouTube 网络分别在 $\varepsilon=0.55$ 和 $\varepsilon=0.45$ 时取得 *AUC* 的最大值. 对于 Digg 网络，其变化趋势存在两个波峰，并且在最后一个波峰取得最大值. 考虑到这种情况，我们通过研究数据集的采样过程发现，Digg 网络在收集过程中虽然未出现类似 Flickr 以及 YouTube 网络的断点的情况，但是 Digg 网络却存在突变(如表 7)(突然数据在某一个时间点出现大量的新边和新节点)，因此这种突变可能对网络本身的拓扑性质造成较大的影响，因此和一般正常网络的采集所形成的图像变化有所差异.

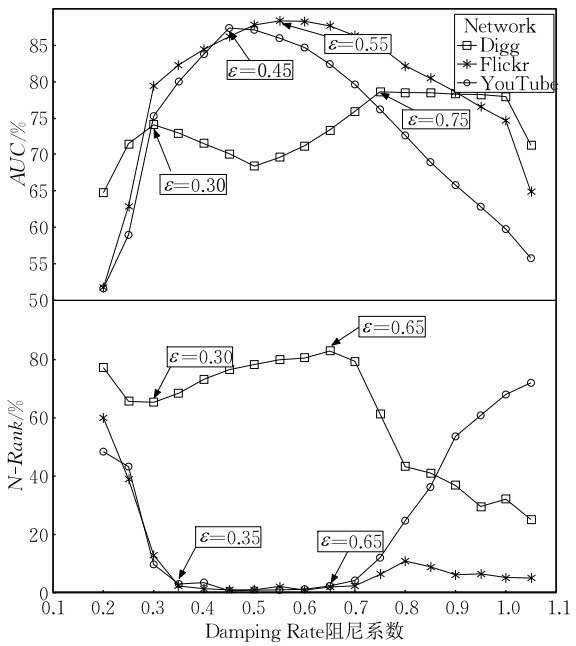


图 9　阻尼系数调参变化

然而，通过实验我们发现：无论是哪种网络，随着 Damping 系数的变化，并不会引起训练时间的改变(1%以内)，因此图 9 中并未显示对时间的影响. 进一步对比网络的拓扑特征后，我们发现，不同网络取得 *AUC* 极值点的阻尼值大小和网络的平均最短路径长度(MSPL)存在关系. 一般的，对于 MSPL 值较小的网络而言，*AUC* 取得最大值所对应的 ε 值通常较小，反之较大. 平均最短路径的作用体现在预测过程中相互影响的平均极大贡献度. 因此当 MSPL 较小时，相比于同样的 ε 值，MSPL 较小网络的平均极大贡献度越大，从而 *AUC* 值越容易取得最大值(或已经取得过最大值). 总之，对于不同的网络而言，在训练过程中，对于超参数 ε 的取值，我们建议根据网络的 MSPL 值来进行选取和调试：当 MSPL 较小时，选取在较小的 ε 值附近进行调试，反之在较大的 ε 值附近进行调试.

与 *AUC* 值的变化趋势不同的是：*N-Rank* 的取值的变化趋势在一定程度上与其是相反的. 例如，对于 YouTube 网络，其 *AUC* 值在 $\varepsilon=0.45$ 时取得最大值，而此时的 *N-Rank* 得分却非常低. 在本文中，*AUC* 得分越高，表示当前学习得到的网络结果能够有效地区分当前不存在的边与下一阶段可能形成的边的存在绝对概率比. 而 *N-Rank* 的得分越高，表示当前学习得到的网络能够有效区分当前不存在的边与下一阶段可能形成的边的存在相对概率比. 根据应用任务的不同，*AUC* 与 *N-Rank* 的重要程度也有所差异. 因此，需要根据具体任务对上述参数进行调整. 通常，*AUC* 值作为主要考虑指标.

(2) 向量维度数 d

参数向量的维度数是目前多个嵌入模型[7,9,12]的重要超参数之一. 在文本词向量的学习中，通常认为参数向量维度数的增加，会带来较大的预测准确率的提升，然而 DNPS 模型中却存在较大差异. 如图 10 所示，随着向量维度数的增加，YouTube 网络与 Digg 网络的动态链接预测 *AUC* 和 *N-Rank* 值都发生了较大的变化(降低或者提升). 其中，Digg 网

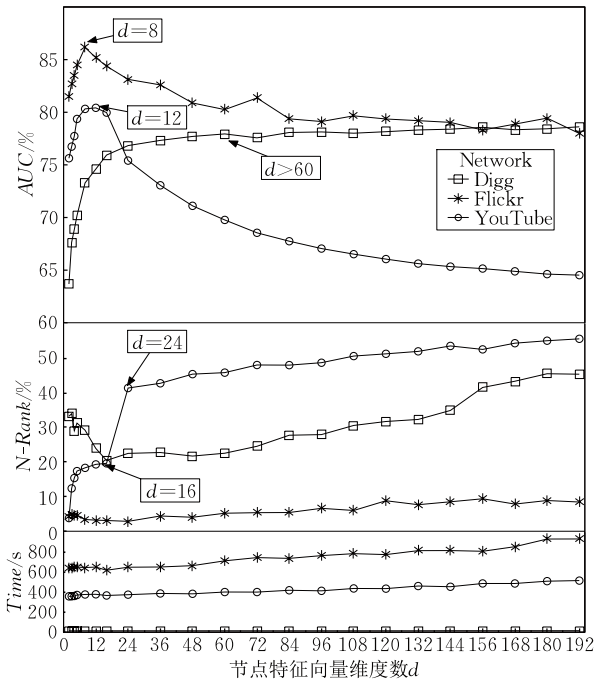


图 10　节点向量维度数调参

络的 AUC 值对于网络向量的维度数的反应最大，当 d 从 2 维增加到 24 维时，其 AUC 预测的准确率发生较大提升(+13.15%)，而后，随着维度数的增加，AUC 值增速并不明显，并且趋于稳定. 而 Digg 网络对应的 N-$rank$ 值，在经历了小幅度的下降后，同样随着维度数的增加而增加.

相对于其它网络而言，YouTube 网络对于向量维度数的变化所造成的 N-$Rank$ 影响则并不明显，这是由于 YouTube 网络本身的稀疏性造成的(YouTube，Flickr，Digg 网络的平均集聚度系数分别为 0.138%，10.8%，6.14%). 在训练过程中，由于网络的稀疏性较高，通常使得网络的特征分布更为分散，因此，网络维度数的变化不易捕获不存在边与未来存在的边之间的相对特征. 相反，过多的增加维度数，反而会使得绝对特征的概率小于相对特征的概率，从而导致 AUC 值在较大的维度数时，反而更低. 同时，还可以发现：维度数的增加，对于网络训练时间的影响都是正向的，但也非指数级增长，因此，在增加维度数的同时，还需要考虑因此而带来的额外时间开销.

总体上来说，在网络的特征提取过程中，不同于文本向量，应该选取在较低的向量维度附近进行超参实验. 较低的，适当的维度设置，不仅仅有利于获取较高的预测准确率，还有利于降低模型的训练时间开销与内存开销.

(3) 节点最低增长率 λ

由于在阻尼正采样过程中使用随机采样的方式，因此，每次随机路径采样所覆盖的节点将会存在差异，这种差异的个数与节点总数的比值即为节点的增长率. 如果当前节点的增长率小于超参数(节点最低增长率 λ)时，本次训练的正采样过程结束. 因此可以看到，对于固定的节点总数，节点最低增长率 λ 值越小，对于同一个网络，需要的可能随机路径采样次数也就越多，此时，网络特征提取与模型训练时间也就越长.

如图 11 所示，为节点最低增长率的调参变化曲线. 模型中，可以看到 YouTube 网络的在调参训练过程中并未发生变化，通过对 YouTube 网络进行分析后发现，由于 YouTube 网络较为稀疏，当使用 DNPS 的采样算法时，只需要一次扫描过程，即可覆盖全部的网络节点，从而使得 λ 的改变并不会引起较大的节点覆盖路径的变化. 相比于 YouTube 网络，Flickr 网络以及 Digg 网络则对 λ 系数的变更较为敏感.

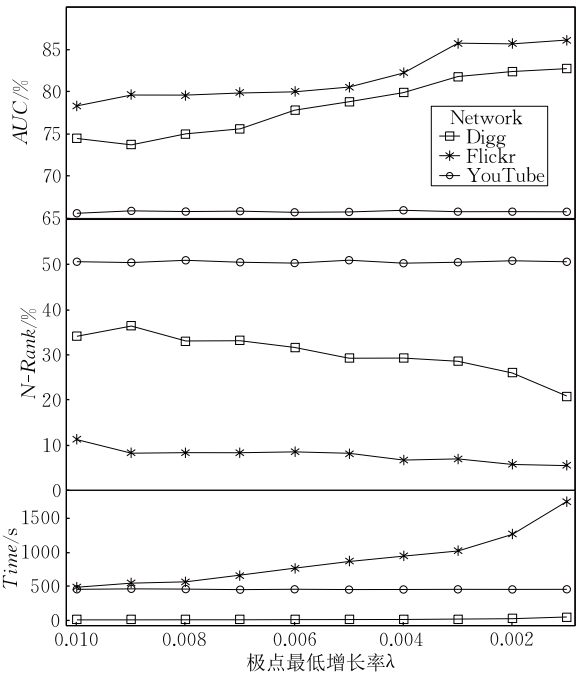


图 11 节点最低增长率调参

通过图 11 可以发现，随着 λ 系数减小，使得训练过程中，单次扫描的最低终止条件更加苛刻，从而增加了网络的路径覆盖度，使得网络的训练更加充分(AUC 值升高)，但同时带来了训练时间的快速增长. 因此，在实际的训练过程中，需要综合考虑模型的时间开销等需求，对 λ 参数进行调整.

(4) 节点负采样个数

如图 12 所示，为 DNPS 模型中每次预测时所匹配负采样节点个数的调参变化曲线. 在模型的训练过程中，对于每一次正样例预测，DNPS 模型都会通过在负采样表中查找，匹配以一定数量(即超参数 $|N_x|$)的负样例数，用以训练. 然而，如图 12 所示，对于不同类型的网络，网络取得最优 AUC 结果的负样例个数存在较大的差异. 其中，由于网络的稀疏性较大，YouTube 网络相比于 Flickr 网络和 Digg 网络更容易受到负样例数的影响. 对比 Flickr 网络和 Digg 网络的变化曲线可以发现，虽然 DNPS 模型在负样例的选取过程中已经采用了剪枝策略(第 4.3 节)，但由于 Flickr 网络的平均度远大于 Digg 网络，因此，当持续增加负样例个数时，仍然有较大的可能性选择到正样例(即错误的把正样例当做负样例进行训练. 因此，对于有向网络而言，当网络的聚集度或者平均度较大时，建议选择较小的 $|N_x|$ 超参数进行训练，反之，则可以适当增大 $|N_x|$ 数目. 同时还应该注意到，对于上述 3 种网络，增大 $|N_x|$ 数目，都会直接导致训练之间线性提高，因此在选择合

适的$|N_x|$数目的同时，还应该考虑模型训练的整体时间开销.

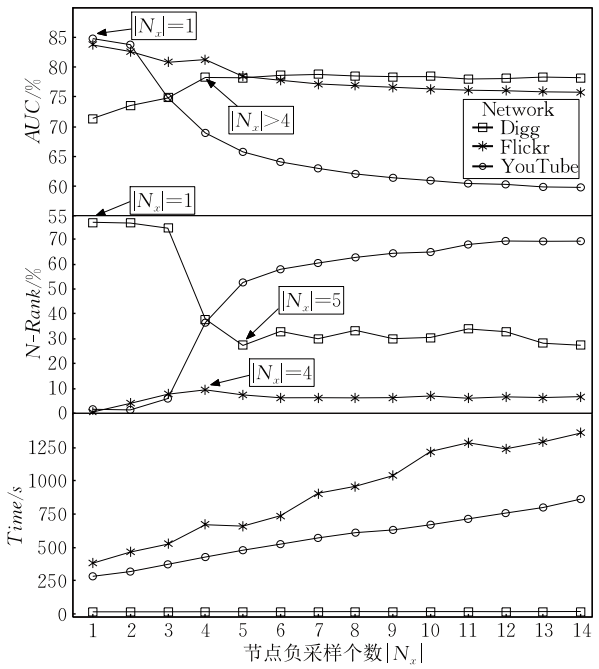


图 12　节点负采样个数调参

超参数选取与总结. 本小节主要对 DNPS 模型中所涉及到的 4 个主要超参数：阻尼系数、向量维度数、节点最小增长率以及负样例匹配个数进行影响结果分析. 实验表明，上述超参在不同的网络中所造成的影响具有较大的差异，同时 *AUC* 与 *N-Rank* 值在不同的变化阶段也存在差异. 其中，网络的集聚度系数以及网络的平均最短路径是影响模型特征提取时超参数选取的主要考虑因素. 因此，对于特定的动态社会网络，需要根据网络的拓扑结构特征，在建议的参数选取范围内进行调参，以获取网络结构特征提取的最佳效果.

6　结论与未来研究

本文主要研究了基于大规模动态社会网络的环境下节点结构特征的表示学习问题. 通过对不同阶层的节点关系进行正负阻尼采样，同时构建针对新增网络节点的局部搜索动态采样算法，提出了 DNPS 模型. 和最新的网络特征学习模型 DeepWalk 以及 LINE 模型的对比实验表明：DNPS 模型在大规模动态社会网络上具有良好的特征提取能力，同时其时间开销得到了有效的控制. DNPS 模型的提出解决了大规模动态社会网络分布式特征学习过程中存在的动态性问题以及社会网络所具备的层次关系的模拟问题. 最后，我们通过建立项目网站①为研究社区提供了 DNPS 模型训练所需的代码以及友好的参考文档，为 DNPS 模型在相关领域的应用，以及后续算法的对比提供参考. 网络表示学习是近两年内，复杂网络领域的新兴研究方向之一. 学习得到良好的网络特征表示，不仅能够解决目前大规模动态社会网络所存在的特征稀疏性以及动态性等问题. 还能有效的应用到其它相关子领域，例如大规模动态社区发现[25]，大规模社会网络中的用户分类[26]以及大规模社会网络信息中的信息扩散[27]等.

针对本文模型而言，为了简化，此处提出的是基于全局的一致的阻尼基准值. 进一步的可能研究方向有：(1) 基于节点关系，构建动态阻尼模型，即采用类似 LSTM(Long-Short Term Memory)模型中的“对齐”思想，针对每次正采样构建“对齐”的动态阻尼值；(2) 本文所提出的超参数稍显冗余，如何针对不同的网络类型进一步减少超参数的个数，减少应用者的使用难度，提升模型的训练效果.

超出本文模型的研究方向还可能有：(1) 跨媒体网络的特征学习. 在同一个社会网络中，不仅仅存在节点之间的结构特征，还有可能存在节点属性，社区属性以及边属性等多种媒体类型，因此，如何有效的对不同媒体的特征基于节点本身进行整合，得到更加丰富的网络特征表现，值得后续进一步探讨和分析；(2) 跨平台网络的特征学习. 在单一网络平台特征得到有效学习的基础上，对于不同网络平台之间的特征进行学习与对应. 例如，用户在新浪微博上存在网络关系的同时，亦可能在其它社会网络平台上存在类似关系. 那么，是否同一个用户节点在不同的社会网络平台上所存在的(或学习得到的)特征是一致的呢？特别是在大规模的环境下，这种跨平台的一致性对应研究，能够有效的挖掘用户之间的关联，可能成为跨平台推荐预测的有效方法之一.

参　考　文　献

[1] Wang P, Xu B W, Wu Y R, et al. Link prediction in social networks: The state-of-the-art. Science China Information Sciences, 2015, 58(1): 1-38

[2] Aouay S, Jamoussi S, Gargouri F, et al. Modeling dynamics of social networks: A survey//Proceedings of the International

① http://www.n2v.org/cjc_2016li.html

Conference on Computational Aspects of Social Networks. Porto, Portugal, 2014: 49-54

[3] Xu Ke, Zhang Sai, Chen Hao, et al. Measurement and analysis of online social networks. Chinese Journal of Computers, 2014, 37(1): 165-188(in Chinese)
(徐恪, 张赛, 陈昊等. 在线社会网络的测量与分析. 计算机学报, 2014, 37(1): 165-188)

[4] Bengio Y, Courville A, Vincent P. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(8): 1798-1828

[5] Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch. The Journal of Machine Learning Research, 2011, 12: 2493-2537

[6] 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, Stroudsburg, USA, 2012: 873-882

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

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

[9] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: Online learning of social representations//Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA, 2014: 701-710

[10] Cao S, Lu W, Xu Q. GraRep: Learning graph representations with global structural information//Proceedings of the 24th ACM International Conference on Information and Knowledge Management. New York, USA, 2015: 891-900

[11] Zhiyuli A, Liang X, Zhou X P. Learning structural features of nodes in large-scale networks for link prediction//Proceedings of the 30th AAAI Conference on Artificial Intelligence. Phoenix, USA, 2016: 4286-4287

[12] Tang J, Qu M, Wang M, et al. LINE: Large-scale information network embedding//Proceedings of the 24th International Conference on World Wide Web. New York, USA, 2015: 1067-1077

[13] Li Zhi-Yu, Liang Xun, Zhou Xiao-Ping, et al. A link prediction method for large-scale networks. Chinese Journal of Computers, 2016, 39(10): 1947-1964(in Chinese)
(李志宇, 梁循, 周小平等. 一种大规模网络中基于节点结构特征映射的链接预测方法. 计算机学报, 2016, 39(10): 1947-1964)

[14] Tenenbaum J B. A global geometric framework for nonlinear dimensionality reduction. Science, 2000, 290(5500): 2319-2323

[15] Roweis S T. Nonlinear dimensionality reduction by locally linear embedding. Science, 2000, 290(5500): 2323-2326

[16] Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering//Proceedings of the 2011 Conference on Neural Information Processing Systems. Vancouver, Canada, 2001: 585-591

[17] Teh Y W, Roweis S T. Automatic alignment of hidden representations//Proceedings in Advances in Neural Information Processing System. Cambridge, USA, 2002, 15: 841-848

[18] Li X Y, Du N, Li H, et al. A deep learning approach to link prediction in dynamic networks//Proceedings of the 2014 SIAM International Conference on Data Mining. Philadelphia, USA, 2014: 289-297

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

[20] 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

[21] 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

[22] Ahn Y-Y, Bagrow J P, Lehmann S. Link communities reveal multiscale complexity in networks. Nature, 2010, 466(7307): 761-764

[23] Cheng Qing, Huang Sen, Huang Jin-Cai. Hierarchical structure discovery in social nNetworks. Complex Systems and Complexity Science, 2015, 12(1): 8-16(in Chinese)
(成清, 黄森, 黄金才. 社会网络的层次结构发现. 复杂系统与复杂性科学, 2015, 12(1): 8-16)

[24] Gutmann M U, Hyvärinen A, Gutmann M U, et al. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. The Journal of Machine Learning Research, 2012, 13(1): 307-361

[25] Chen Yu-Zhong, Shi Song, Zhu Wei-Ping, et al. An incremental community discovery algorithm based on neighborhood following relationship, 2017, 40(3): 570-583(in Chinese)
(陈羽中, 施松, 朱伟平等. 一种基于邻域跟随关系的增量社区发现算法. 计算机学报, 2017, 40(3): 570-583)

[26] Tuarob S, Tucker C S. Automated discovery of lead users and latent product features by mining large-scale social media networks. Journal of Mechanical Design, 2015, 137(7): 071402

[27] Lerman K, Ghosh R. Information contagion: An empirical study of the spread of news on Digg and Twitter social networks//Proceedings of the 4th International AAAI Conference on Weblogs and Social Media. Washington, USA, 2010: 90-97

LI Zhi-Yu, born in 1991, Ph. D. candidate, a member of the Outstanding Innovative Talents Cultivation Funded Programs of RUC. His research interests include social computing and 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.

XU Zhi-Ming, born in 1993, M. S. candidate. His research interests include social computing and parallel computing.

QI Jin-Shan, born in 1978, Ph. D. candidate. His research interests include social networks and user identification.

CHEN Yan-Fang, born in 1992, Ph. D. candidate. Her research interests include social network and data mining.

Background

Network representation learning is becoming a hot topic recently in network modeling research area. Most of the past works are based on the assumptions that the networks are static and small. The model or algorithm is designed very complicated and time explosion, which makes the excessively long processing times and excessively large memory space requirement become major problems for large-scale networks modeling nowadays.

As an important component parts for networks, social networks are different from the language networks, power networks or any other small networks. The lability and giant are two key features of social networks. It's hard to learning the distributed representation of social networks with the exist network embedding methods.

In this paper, we propose a method namely DNPS, to handle the problem of learning distributed representation of large-scale dynamic social networks. We first propose a damping based positive sampling algorithm to generate the positive training sets, which can satisfy the features of social network that the nodes are hierarchical. With this method, the weight of co-occurrence between nodes is various acoording to the ranks of them. Considering the fact that social networks are extremely labile, and the network evolution is a common phenomenon of most existent social networks. So, learning the distributed representation of dynamic social networks has become challenging to deal with. To solve this problem, we proposed an incremental learning algorithm based on local search around the newly added nodes or links to generate positive pairs from evolved networks.

At the end of this paper, we conduct extensive experimental analysis on three large-scale dynamic social networks with the task of dynamic future links prediction. We compared our model with two famous network-embedding methods, LINE and DeepWalk, both in time and AUC test. The experiments show that our method outperforms the LINE and DeepWalk methods in several experiment settings. Especially the dynamic prediction tests show that DNPS method is good at handling the problem of learning distributed representation for large-scale dynamic social network in a high efficient working style.

The codes are available at http://www.n2v.org/cjc_2016li.html.

This work is partly supported by the National Natural Science Foundation of China (Grant Nos. 71271211 and 71531012), the Fundamental Research Funds for the Central Universities (the Research Funds of Renmin University of China, Grant No. 10XNI029), the Natural Science Foundation of Beijing (Grant No. 4172032), and the Outstanding Innovative Talents Cultivation Funded Programs 2016 of Renmin University of China.