

文章编号: 1000-5641(2020)05-0083-12

# 网络顶点表示学习方法

周晓旭<sup>1</sup>, 刘迎风<sup>2</sup>, 付英男<sup>1</sup>, 朱仁煜<sup>1</sup>, 高明<sup>1</sup>

(1. 华东师范大学 数据科学与工程学院, 上海 200062; 2. 上海市大数据中心, 上海 200072)

**摘要:** 网络是一种常用的数据结构, 在社交、通信和生物等领域广泛存在, 如何对网络顶点进行表示是学术界和工业界广泛关注的难点问题之一. 网络顶点表示学习旨在将顶点映射到一个低维的向量空间, 并且能够保留网络中顶点间的拓扑结构. 本文在分析网络顶点表示学习的动机与挑战的基础上, 对目前网络顶点表示学习的主流方法进行了详细分析与比较, 主要包括基于矩阵分解、基于随机游走和基于深度学习的方法, 最后介绍了衡量网络顶点表示性能的方法.

**关键词:** 网络嵌入; 随机游走; 矩阵分解; 深度神经网络

**中图分类号:** TP391 **文献标志码:** A **DOI:** 10.3969/j.issn.1000-5641.202091007

## Approaches on network vertex embedding

ZHOU Xiaoxu<sup>1</sup>, LIU Yingfeng<sup>2</sup>, FU Yingnan<sup>1</sup>, ZHU Renyu<sup>1</sup>, GAO Ming<sup>1</sup>

(1. School of Data Science and Engineering, East China Normal University, Shanghai 200062, China;

2. Shanghai Municipal Big Data Center, Shanghai 200072, China)

**Abstract:** Network is a commonly used data structure, which is widely applied in social network, communication and biological fields. Thus, how to represent network vertices is one of the difficult problems that is widely concerned in academia and industry. Network vertex representation aims at learning to map each vertex into a vector in a low-dimensional space, and simultaneously preserving the topology structure between vertices in the network. Based on the analysis of the motivation and challenges of network vertex representation, this paper analyzes and compares the mainstream methods of network vertex representation in detail, including matrix decomposition, random walk and deep learning based approaches, and finally introduces the methods to measure the performance of network vertex representation.

**Keywords:** network embedding; random walk; matrix factorization; deep neural network

## 0 引 言

现实世界中普遍存在类型丰富多样的网络数据结构, 如社交网络<sup>[1]</sup>、通信网络、生物网络<sup>[2]</sup>等, 这些网络表示实体之间的关系, 规模从数百个顶点到数百万个甚至数十亿个顶点<sup>[3]</sup>. 分析网络在许多应用中都发挥着至关重要的作用, 因此也得到学术界和工业界越来越多的关注. 对网络进行有效分析首

收稿日期: 2020-08-05

基金项目: 国家重点研发计划 (2016YFB1000905); 国家自然科学基金 (U1911203, U1811264, 61877018, 61672234, 61672384); 中央高校基本科研业务费专项; 上海市科技兴农推广项目 (T20170303); 上海市核心数学与实践重点实验室资助项目 (18dz2271000)

通信作者: 刘迎风, 男, 副高级工程师, 研究方向为数据运营、数据分析和用户画像. E-mail: yfliu@shanghai.gov.cn

先要对网络顶点进行表示. 邻接矩阵、拉普拉斯矩阵等传统表示方法展示了顶点间的显式关系, 更复杂的、高阶的拓扑结构很难被有效地表示出来, 而且这类传统方法通常复杂度较高, 难以应用在大规模网络上. 为了突破传统方法的“瓶颈”, 顶点嵌入技术旨在在尽可能保留网络拓扑结构、顶点语义信息和顶点属性的前提下, 将顶点映射到低维向量空间中, 该空间也被称之为顶点嵌入空间. 嵌入空间不仅能够重构原始网络, 而且保留了网络的拓扑特征. 如果两个顶点在网络拓扑结构上是相似的, 那么这两个顶点在嵌入空间中也要尽量靠近.

在顶点嵌入后, 可以利用机器学习或者深度学习方法在嵌入空间中高效地进行网络分析, 如链接预测<sup>[4]</sup>、顶点分类<sup>[5]</sup>和网络可视化<sup>[6]</sup>. 链接预测指预测缺失的链接或将来可能产生的链接. 顶点分类指基于有标记的顶点和网络拓扑关系进一步确定其他顶点的标签. 网络可视化将原始网络降维到二维空间中展示, 有助于直观理解网络中顶点间的相互关系.

## 1 问题与挑战

由于网络数据结构存在一些独特的特征, 致使网络顶点表示学习面临众多问题和挑战, 其中三个最关键挑战分别是:

(1) 稀疏性 真实世界中的网络顶点的度通常服从长尾分布, 能够观察到的边仅占很小的比例, 许多顶点间的边存在缺失, 实际上大多数顶点仅有很少的连接, 少数顶点拥有更多的连接. 由于稀疏的网络链接给顶点表示造成了困难, 因此网络中普遍存在的稀疏特征给顶点表示学习带来了很大的挑战.

(2) 保留结构和属性 顶点表示学习需要保留原始网络中的局部结构和全局结构. 网络中彼此相近的顶点在嵌入空间中也应该尽量靠近, 顶点间的相对位置也要保持不变. 此外, 网络顶点的属性信息也是网络的重要组成部分, 但如何将这些信息与网络拓扑结构综合在一起? 因此, 同时保留顶点的结构和属性信息是网络顶点嵌入学习的重要挑战.

(3) 可扩展性 现实世界中的信息网络其规模越来越庞大, 从数百个顶点到数百万个顶点甚至数十亿个顶点, 分析大规模信息网络的需求越来越迫切, 但也要避免耗费昂贵的计算代价. 因此, 网络表示学习方法要具备可扩展性, 能够针对大规模网络进行顶点表示学习, 并进一步支持针对大规模网络的其他分析与应用.

## 2 问题定义

**定义 1(网络)** 网络  $G(V, E)$  由顶点集合  $V = \{v_1, \dots, v_n\}$  和边集合  $E = \{e_{ij}\}$  构成的二元组, 其中  $e_{ij}$  表示  $v_i$  与  $v_j$  之间的连接. 无权网络  $G$  的邻接矩阵记为  $A$ , 其中当  $v_i$  与  $v_j$  存在连接时,  $a_{ij}=1$ , 否则  $a_{ij}=0$ . 对于无向网络  $G$ ,  $A$  具有对称性, 即  $a_{ij}=a_{ji}$ . 带权网络  $G$  的权重矩阵记为  $S$ , 其中  $s_{ij}$  表示  $v_i$  与  $v_j$  连接上的权重. 权重能够表示顶点间联系的强弱.

网络顶点在嵌入低维向量空间的过程中, 需要利用顶点间的局部相似性. 在 LINE<sup>[3]</sup> 中提出了顶点间的局部结构相似性.

**定义 2(一阶相似性)** 一阶相似性表示网络中两个顶点间的局部相似性. 如果顶点  $v_i$  和  $v_j$  间有边连接, 则边上的权重  $s_{ij}$  即为顶点  $v_i$  和  $v_j$  的一阶相似度; 若没有边连接,  $s_{ij} = 0$ , 则一阶相似性也为 0.

在现实网络中, 能够观察到的边仅占很小的比例, 而许多顶点间的边都缺失了, 这种顶点对的一阶相似性为 0, 一阶相似性忽略了顶点间的隐式关系. 因此为了保留更丰富的网络结构, 进一步定义顶点之间的高阶相似性.

**定义 3(二阶相似性)** 二阶相似性表示网络中两个顶点之间的邻居相似性. 用  $p_i = (w_{i,1}, \dots,$

$w_{i,|V|}$ 表示顶点  $v_i$  与其他顶点的一阶接近度,  $w_{i,j}$ 表示顶点  $v_i$  和顶点  $v_j$  的一阶相似性, 二阶相似性可以通过  $p_i$  和  $p_j$  的相似性确定. 若顶点  $v_i$  与  $v_j$  间不存在一样的邻居顶点, 则二阶相似性为 0.

二阶相似性考虑两个顶点邻域间的相似性, 如果它们具有相似的邻居, 则认为它们是相似的. 在社交网络中<sup>[1]</sup>, 有相同朋友的人往往具有相似的兴趣, 很可能也会成为朋友. 在单词共现网络中, 经常与同一组单词同时出现的单词往往具有相似的含义.

如图 1 所示, 顶点 6 和顶点 7 之间存在边, 且权重较大, 则认为两者一阶相似性较高. 而顶点 5 和顶点 6 之间不存在边, 则两者间的一阶相似性为 0, 但是由于它们的邻居重合度较高, 则认为两者的二阶相似性较高.

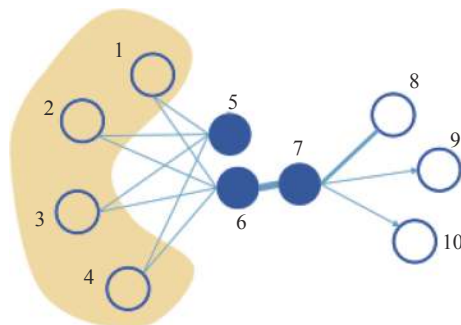


图 1 网络一阶、二阶相似性示例<sup>[3]</sup>

Fig. 1 An example of first-order proximity and second-order proximity

**定义 4(网络嵌入)** 给定网络  $G(V, E)$ , 网络顶点嵌入通过特定的映射函数  $f$  将网络中的每个顶点转换为低维嵌入向量空间中的潜在表示  $f: v \mapsto y \in \mathbb{R}^d$ , 其中  $d$  为嵌入向量空间的维度, 其值远小于网络中顶点集合的大小  $|V|$ .

学习到的顶点表示空间需要保留网络结构并可以重构原始的网络. 如果两个顶点之间存在链接, 为了保留网络关系, 这两个顶点在嵌入空间中的距离也应相对较小.

### 3 网络顶点表示学习方法的分类

长久以来, 针对网络顶点表示学习已经提出了许多方法. 本文根据它们在图的顶点表示方法上的差异, 将这些模型分成三大类别: 基于矩阵分解的方法、基于随机游走的方法和基于神经网络的方法. 接下来介绍每一类里的代表性方法.

#### 3.1 基于矩阵分解的方法

对于有  $n$  个顶点的网络  $G$ , 用  $n \times n$  的邻接矩阵  $A$  来表示网络的拓扑结构, 每一行每一列的值代表顶点和其余顶点的连接关系, 值为 0 表示顶点之间不存在边. 邻接矩阵的行向量或列向量构成了顶点的一种  $n$  维表示, 这种表示的维度通常都很高, 而网络嵌入想要得到的是低维顶点向量表示. 通过矩阵分解、维度约减, 可以将高维的原始矩阵分解获得低维表示<sup>[7-8]</sup>. 例如, 对邻接矩阵、拉普拉斯矩阵、转移概率矩阵等进行特征值分解、奇异值分解、非负矩阵分解, 从而获得顶点的低维表示.

##### 3.1.1 Locally Linear Embedding

局部线性嵌入 (Locally Linear Embedding, LLE)<sup>[7]</sup> 假设顶点可以由它的邻居向量线性表示, 降维之后仍能保持同样的线性关系, 即权重系数基本保持不变.

LLE 首先利用 K-最近邻算法或其他方法确定降维前顶点  $v_i$  的  $k$  个邻居顶点, 然后使用最小化均方误差找到线性表示的归一化权重系数  $w_{ij}$ ,  $w_{ij}$  表示顶点  $v_j$  对于重构顶点  $v_i$  的重要性. 即对于所有顶点:

$$\min_W \sum_i \|v_i - \sum_{j \in n(i)} w_{ij} v_j\|^2, \text{ 其中 } \sum_j w_{ij} = 1. \quad (1)$$

由于在嵌入空间中仍然要保持顶点间的局部线性关系, 因此 LLE 利用第二步得到的权重系数重构顶点和邻居间的线性关系, 通过最小化均方误差损失函数能够得到顶点的低维表示:

$$\min_i \sum_i \|y_i - \sum_{j \in n(i)} w_{ij} y_j\|^2. \quad (2)$$

LLE 算法对最近邻居顶点的选择较为敏感, 选择不同的最近邻居顶点会影响降维结果, 并最终影响顶点的低维表示.

### 3.1.2 Laplacian Eigenmaps

拉普拉斯特征映射 (Laplacian Eigenmaps, LE)<sup>[8]</sup> 的思路是如果边的权重高, 那么在降维后的嵌入空间中这两个顶点的表示也尽量靠近. 该方法主要有 3 个步骤: 首先, 利用 KNN 算法或者  $\epsilon$ -邻域方法构建图, 将邻近的顶点用边连接起来; 然后, 确定边上的权重, 可以利用热核函数确定权重  $w_{ij}$ , 或简单地让  $w_{ij}$  为 1 或 0; 最后, 最小化目标函数

$$\min_{i,j} \sum_{i,j} \|y_i - y_j\|^2 w_{ij} = \text{trace}(Y^T L Y). \quad (3)$$

其中拉普拉斯矩阵  $L = \Lambda - W$ , 度矩阵中对角元的值  $\Lambda_{ii} = \sum_j w_{ij}$ , 非对角元为 0. 该问题转化为拉普拉斯矩阵  $L$  的非零特征值求解问题, 对应的特征向量就是顶点的嵌入表示. 然而, LE 的时间复杂度是顶点数量的平方, 对于具有百万顶点的网络这种方法并不适用, 缺乏可扩展性.

### 3.1.3 GraRep

与已有工作相比, 除了考虑一阶、二阶相似性, GraRep<sup>[9]</sup> 认为还应捕获更高阶的相似性以保留更丰富的网络结构. 给定邻接矩阵  $A$ , 可以通过  $A^k$  计算  $k$  步概率转移矩阵,  $A_{ij}^k$  表示顶点  $v_i$  从当前经过  $k$  步到达顶点  $v_j$  的转移概率  $p_k(j|i)$ . 利用 Noise Contrastive Estimation 模型和负采样的思想, 来定义损失函数

$$L_k = \sum_{i \in V} L_k(i). \quad (4)$$

其中

$$L_k(i) = \left( \sum_{j \in V} p_k(j|i) \log \sigma(y_j^T y_i) \right) + \lambda E_{j' \sim p_k(V)} (\log \sigma(-y_i^T y_{j'})). \quad (5)$$

然后对矩阵  $X^k \approx X_d^k = W^k C^k$  进行奇异值分解, 其中  $X^k$  的元素为  $X_{i,j}^k = \max(Y_{i,j}^k, 0)$ ,  $W^k = U_d^k (\Sigma_d^k)^{\frac{1}{2}}$ ,  $C^k = (\Sigma_d^k)^{\frac{1}{2}} (V_d^k)^T$ . 经过分解后便可得到顶点的  $k$  步嵌入表示为  $W^k$ . 最后把所有的  $W^k$  连接起来, 可以获得顶点的全局表示. 由于 GraRep 的复杂度为  $O(|V^3|)$ , 因此同样不适用于大型网络.

### 3.1.4 HOPE

传递性在无向图中是对称的, 但在有向图中对称性并不成立. 非对称的传递性是指如果存在从顶点  $u$  到顶点  $w$  和从顶点  $w$  到顶点  $v$  的有向边, 则可能存在从顶点  $u$  指向顶点  $v$  的边, 但没有从  $v$  指向  $u$  的边.

High-Order Proximity preserved Embedding, 简称 HOPE<sup>[10]</sup>, 扩展了已有工作<sup>[11]</sup>, 保留了有向图中的非对称传递性. HOPE 总结的 4 种方法 Kate Index、Rooted PageRank、Common Neighbors 和 Adamic-Adar 都能够以相同的形式衡量相似性矩阵  $S = M_g^{-1} M_l$ , 其中  $M_g$  和  $M_l$  都是稀疏的. 然后使用



广义奇异值分解  $M_g^{-1} M_l = V^s \Sigma (V^t)^T$  获得广义奇异值  $\{\sigma_1^l, \dots, \sigma_d^l\}$ 、 $\{\sigma_1^g, \dots, \sigma_d^g\}$  和对应的奇异向量  $\{v_1^s, \dots, v_d^s\}$ 、 $\{v_1^t, \dots, v_d^t\}$ . 接着根据  $\sigma_i = \frac{\sigma_i^l}{\sigma_i^g}$  计算得到奇异值. 最后按照公式

$$U^s = [\sqrt{\sigma_1} v_1^s, \dots, \sqrt{\sigma_d} v_d^s], U^t = [\sqrt{\sigma_1} v_1^t, \dots, \sqrt{\sigma_d} v_d^t] \quad (6)$$

得到嵌入矩阵  $U = [U^s, U^t]$ .

### 3.2 基于随机游走的方法

基于矩阵分解得到的向量往往是稀疏的, 而且这些方法复杂度较高, 不适合应用在大规模网络和下游任务中. 在自然语言处理领域, Word2Vec<sup>[12-13]</sup> 能够获得单词的连续低维嵌入向量, 它遵循的思路是一个单词的表示可以重构与它相邻的单词表示, 能够捕获自然语言中的语义和语法信息<sup>[14]</sup>. 类似于 Word2Vec 的思想, 在网络上通过随机游走生成随机序列, 将顶点视为单词, 将随机序列视为句子, 产生了基于随机游走的网络嵌入模型.

#### 3.2.1 DeepWalk

DeepWalk<sup>[15]</sup> 是一个基于随机游走学习网络顶点潜在表示的模型, 利用可重复访问已访问顶点的深度优先遍历算法, 它能够保留顶点的邻居结构. DeepWalk 发现, 在随机游走中顶点出现的频数分布和自然语言中的单词分布相似, 都符合幂律分布. 于是借鉴自然语言领域中 Word2Vec 的思路, 利用网络中顶点与邻居的共现关系来学习顶点的潜在表示. 如图 2 所示, DeepWalk 从每个顶点产生长度  $t$  的截断随机游走序列  $s = \{v_{i1}, v_{i2}, \dots, v_{it}\}$ , 再按照 Skip-Gram 方法, 以序列中的每个顶点  $v_i$  为中心, 使它的邻居在大小为  $w$  的窗口中出现的概率最大化, 即最大化如下目标函数:

$$P(\{v_{i-w}, \dots, v_{i+w}\} | v_i) = \prod_{j=i-w, j \neq i}^{i+w} P(v_j | \phi(v_i)). \quad (7)$$

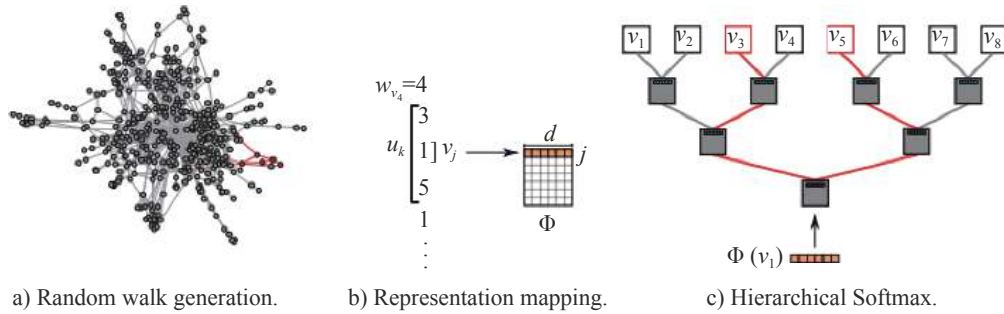


图 2 DeepWalk 算法流程<sup>[15]</sup>

Fig. 2 Workflow for the DeepWalk algorithm

为了加快训练速度, 可用 Hierarchical Softmax 来近似概率分布  $P(v_j | \phi(v_i)) = \prod_{l=1}^{\lceil \log |V| \rceil} P(b_l | \phi(v_i))$ , 其中  $P(b_l | \phi(v_i)) = 1 / (1 + e^{-\phi(v_i) \psi(b_l)})$ ,  $(b_0, b_1, \dots, b_{\lceil \log |V| \rceil})$  是霍夫曼树上以  $v_i$  为根节点的序列. DeepWalk 使用深度优先搜索 (DFS) 策略进行探索, 根据二阶相似性高的顶点产生相似的低维表示来保留二阶相似性, 并没有提出明确的目标函数说明应该保留顶点的哪些属性. 而且, DeepWalk 仅适用于未加权网络.

#### 3.2.2 LINE

LINE<sup>[3]</sup> 可以用于多种类型的大规模网络上的顶点表示, 其中包括无向网络、有向网络、无权网络、有权网络, 而且定义了一阶相似性和二阶相似性. 一阶相似性表示顶点对之间的相似性, 可以用联合概率分布衡量顶点  $v_i$  和  $v_j$  之间的一阶相似性:

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-y_i^T \cdot y_j)}. \quad (8)$$

LINE 的一阶相似性目标函数如式 (9), 即通过最小化与经验分布  $\hat{p}_1(v_i, v_j)$  之间的距离, 使得到的嵌入保留一阶相似性.

$$O_1 = d(\hat{p}_1, p_1). \quad (9)$$

式 (9) 中  $d(\cdot, \cdot)$  用于计算两个分布间的距离, 通常选择 KL 散度 (Kullback-Leibler Divergence) 来度量距离.

二阶相似性认为共享许多邻居的顶点之间是相似的, 用顶点  $v_i$  和  $v_j$  之间的条件分布来衡量:

$$p_2(v_j|v_i) = \frac{\exp(y_j^T \cdot y_i)}{\sum_{k=1}^{|V|} \exp(y_k^T \cdot y_i)}. \quad (10)$$

同样用 KL 散度衡量它与经验分布  $\hat{p}_2(v_i, v_j)$  的差异, 得到 LINE 的二阶相似性目标函数:

$$O_2 = - \sum_{(i,j) \in E} w_{ij} \log p_2(v_j|v_i). \quad (11)$$

通过最小化  $O_1$  和  $O_2$ , LINE 可以得到能够保留一阶和二阶相似性的顶点嵌入.

### 3.2.3 Node2Vec

网络中的顶点有两种重要的相似性结构: 同构性和结构等价性<sup>[16]</sup>. 如图 3 所示, 按照广度优先搜索 (BFS) 策略采样序列, 观察节点周围的微观视图, 能让具有结构等价性的节点在嵌入空间中更接近. 深度优先搜索 (DFS) 策略可以探索网络中距离源节点更远的部分, 采样的节点能够代表邻居的宏观视图, 产生反映同构性的节点潜在表示.

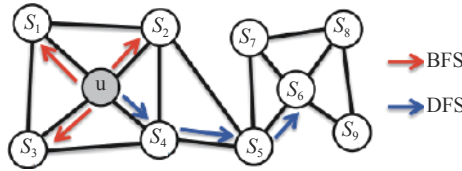


图 3 网络的两种搜索策略<sup>[17]</sup>

Fig. 3 BFS and DFS search strategies

DeepWalk 按照深度优先搜索的方式在网络上进行随机游走, 不足以适应网络中连接模式的多样性. Node2Vec<sup>[17]</sup> 灵活地定义了节点邻居的概念, 提出了有偏的随机游走, 可以兼顾 BFS 和 DFS, 对于具有同构性或者结构等价性的顶点可以学习到相似的嵌入向量.

假定游走中第  $i-1$  个顶点  $c_{i-1}$  为顶点  $v$ , 下一步采样顶点  $x$  的概率为:

$$P(c_i = x | c_{i-1} = v) = \begin{cases} \frac{\pi_{vx}}{Z}, & \text{若 } (v, x) \in E, \\ 0, & \text{其他情况,} \end{cases} \quad \alpha_{pq}(t, x) = \begin{cases} \frac{1}{p}, & \text{若 } d_{tx} = 0, \\ 1, & \text{若 } d_{tx} = 1, \\ \frac{1}{p}, & \text{若 } d_{tx} = 2. \end{cases} \quad (12)$$

其中  $\pi_{vx} = \alpha_{pq}(t, x)w_{vx}$  是未归一化的转移概率,  $Z$  是归一化常数. 引入超参数  $p$  和  $q$  分别控制再一次访问已经访问过的顶点和向外游走或向内游走的概率. Node2Vec 和 DeepWalk 基本都遵循具有相似邻居的节点应具有相似的潜在表示这一原则. 但是, 邻域是由相似性确定的局部概念, 所以对于结构相似但距离远的顶点, 模型无法捕捉它们的结构等价性, 导致在相应任务上表现不佳.

### 3.2.4 Struc2Vec

DeepWalk 和 Node2Vec 在分类任务中表现较好, 但是在结构等价性的任务中往往表现较差. 这是

因为这两种方法采样的序列长度有限, 对相距较远但邻居结构相似的两个节点将不足以学习到相似的潜在表示. 因此, 这两种模型无法正确捕捉结构等价性. 类似于 Subgraph2Vec<sup>[18]</sup>, Struc2Vec<sup>[19]</sup> 更关注网络的结构性, 而且不需要借助节点或边的属性, 依靠节点的度衡量多层次的结构相似性. 如图 4 所示, 顶点  $u$  和  $v$  结构相似距离远, Struc2Vec 能保存它们的结构特征.

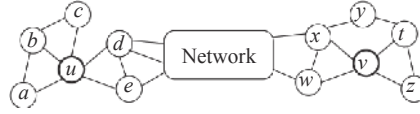


图 4 相距远且结构相似的顶点  $u$  和  $v$  示例<sup>[19]</sup>

Fig. 4 Nodes  $u$  and  $v$  are structurally similar but far apart in the network

首先, Struc2Vec 认为如果两个节点具有相同的度, 那么这两个节点是结构相似的, 而且如果这两个节点的邻居也具有相同的度, 那这两个节点在结构上就会更相似. Struc2Vec 在不同尺寸的邻域和各个层次级别上衡量顶点对的结构相似性.  $R_k(u)$  表示顶点  $u$  经过  $k$  步到达的顶点集合,  $s(V)$  表示对顶点集合  $V$  按度排列的有序序列,  $f_k(u, v)$  表示顶点  $u$  和  $v$  之间经过  $k$  步的结构相似性:  $f_k(u, v) = f_{k-1}(u, v) + g(s(R_k(u)), s(R_k(v)))$ , 其中  $g(\cdot, \cdot)$  衡量两个有序序列的距离.

然后构造层次带权图, 每层图对应于测量结构相似性时的层次级别, 并把不同层次的不同节点连接起来, 权重根据前一步的相似性确定. 接着, 在多层图上进行有偏随机游走用于生成节点序列, 从顶点  $u$  到达同一层顶点  $v$  的概率为  $p_k(u, v) = \frac{e^{-f_k(u, v)}}{Z_k(u)}$ , 分母是正则化因子, 到达其他层中相同顶点的概率为  $p_k(u_k, u_{k+1}) = \frac{w(u_k, u_{k+1})}{w(u_k, u_{k+1}) + w(u_k, u_{k-1})}$  和  $p_k(u_k, u_{k-1}) = 1 - p_k(u_k, u_{k+1})$ , 其中  $w(\cdot, \cdot)$  是层间的边权重. 再利用 SkipGram 从产生的邻居序列中学习顶点的潜在表示.

### 3.2.5 PPNE

Property Preserving Network Embedding, 简称 PPNE<sup>[20]</sup>, 通过共同优化结构驱动和属性驱动的目标函数, 获得能够保存顶点属性的网络嵌入, 如图 5 所示. 之前的工作仅考虑网络拓扑信息<sup>[21]</sup>, PPNE 模型将网络节点属性类型也融合到嵌入中, 旨在为节点学习更好的表示向量, 而节点分类或链接预测任务仅用于评估嵌入结果的质量.

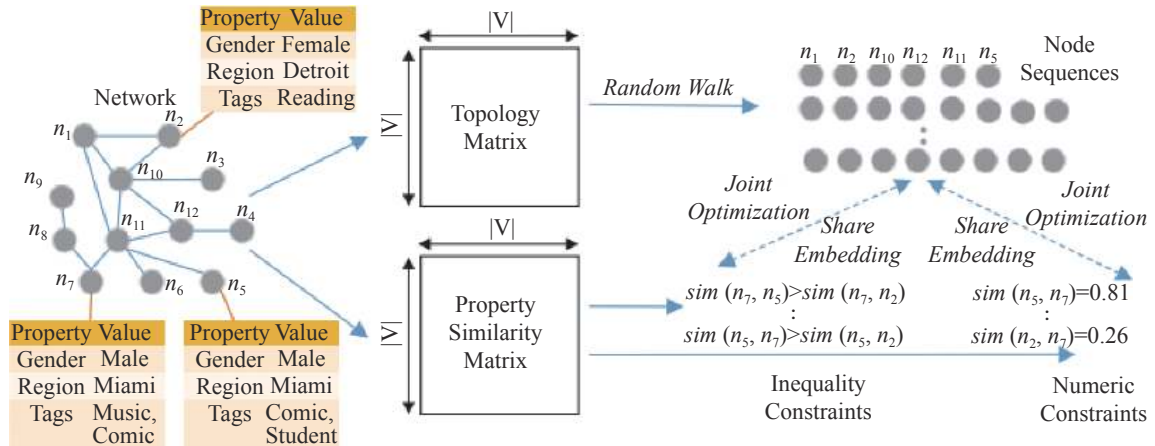


图 5 PPNE 框架<sup>[20]</sup>

Fig. 5 The framework of PPNE

下列优化结构驱动的目标函数 (式 (13)) 使得上下文相似的顶点距离更近.

$$\max D = \prod_{n \in C} \prod_{z \in \text{context}(n)} p(z|n). \quad (13)$$

PPNE 从属性相似性矩阵中提取出不等式约束和数值约束, 基于这些约束构造属性驱动的目标函数:

$$\min D = \sum_{n \in C} \sum_{i \in \{\text{pos}_n \cup \text{neg}_n\}} p_{ni} d(v_n, v_i). \quad (14)$$

其中  $d(v_n, v_i)$  是顶点  $v_n$  和  $v_i$  之间的距离,  $\text{pos}_n$  和  $\text{neg}_n$  是顶点  $n$  的 top- $k$  个相似和不相似的顶点构成的集合.

### 3.3 基于深度学习的方法

网络节点嵌入的学习目标是找到能将原始网络空间转换为低维向量空间的映射函数. 网络的结构复杂, 而且是非线性结构, 因此之前通过线性模型寻找节点嵌入的方法不足以保留信息网络的非线性特征. 而深度神经网络能够对数据中的非线性结构进行建模, 在许多领域都得到了应用且获得了成功. 因此许多工作用端到端的深度网络学习网络顶点表示.

#### 3.3.1 SDNE

与使用浅层模型的嵌入方法相比, 真实网络结构较为复杂, 浅层模型找到的顶点表示向量不能保存网络高阶的非线性特征, 而深度学习擅长捕捉非线性特征. Structural Deep Network Embedding, 简称 SDNE<sup>[22]</sup>, 率先将深度学习应用在网络表示学习中. 与之前使用神经网络对图进行表示的已有工作<sup>[23]</sup>不同, SDNE 学习的是可以在任务之间利用的低维表示, 而且考虑了顶点之间的一阶和二阶相似性. 如图 6 所示, 模型使用半监督学习能够同时保留一阶和二阶网络相似性. SDNE 首先利用无监督学习, 根据顶点的邻居结构, 通过多个非线性函数构成自动编码器获取顶点的潜在表示,  $y_i^{(k)} = \sigma(W^{(k)} y_i^{(k-1)} + b^{(k)})$ , 以此保留网络的二阶相似性. 然后为了保留一阶相似性, 借助拉普拉斯特征映射方法, 有监督地根据邻接矩阵代表的先验知识调整节点的嵌入表示. SDNE 遵循的思路是当相似的顶点在嵌入空间中被映射得较远时, 就让模型的损失变大. 模型的损失函数为:

$$\begin{aligned} L_{\text{mix}} &= L_{2\text{nd}} + \alpha L_{1\text{st}} + v L_{\text{reg}} \\ &= \|(\hat{X} - X) \odot B\|_F^2 + \alpha \sum_{i,j=1}^n a_{ij} \|y_i - y_j\|_2^2 + v L_{\text{reg}}. \end{aligned} \quad (15)$$

其中  $X$ 、 $\hat{X}$  是输入和重构的数据,  $B$  表示邻接矩阵中非零元素的惩罚项系数,  $L_{\text{reg}}$  是防止过拟合的正则化项损失.

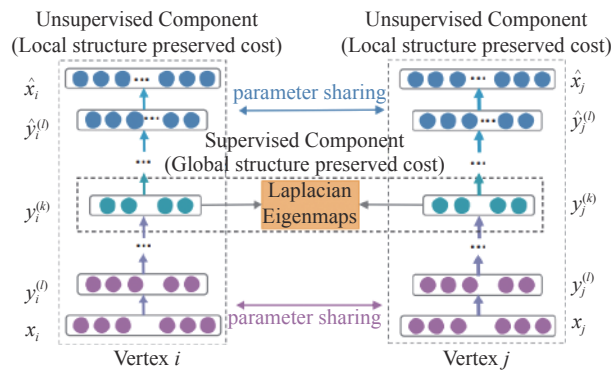


图 6 SDNE 框架<sup>[22]</sup>

Fig. 6 The framework of SDNE

#### 3.3.2 SiNE

符号网络是指具有正边和负边的信息网络, 正边可以表示朋友、信任、喜欢、支持等积极关系, 负



边表示敌人、不信任、讨厌、反对等消极关系. 在各种任务中, 负边包含的价值比正边更多, 因此不能简单地忽视负边的存在<sup>[24]</sup>. 大多数嵌入算法都是为无符号的信息网络设计的. SiNE<sup>[25]</sup> 考虑了符号网络的正负边, 从扩展的结构平衡理论出发, 节点距离它的“朋友”应比“敌人”更近. 对于那些 2 跳网络只有正边的顶点引入虚拟顶点创建负链接, 构成集合  $P = \{(v_i, v_j, v_k) | e_{ij} = 1, e_{ik} = -1, v_i, v_j, v_k \in V\}$  中的三元组. 节点  $v_i$  和  $v_j$  之间的相似度  $f$  应该大于  $v_i$  和  $v_k$  之间的相似度, 即  $f(x_i, x_j) \geq f(x_i, x_k) + \delta$ , 阈值  $\delta$  调节两种相似性之间的距离.  $\delta$  变大会使  $v_i$ 、 $v_j$  更加靠近,  $v_i$ 、 $v_k$  距离更远. SiNE 确定了学习符号网络顶点嵌入的目标函数:

$$\begin{aligned} \min_{X, x_0, \theta} \frac{1}{C} & \left[ \sum_{(x_i, x_j, x_k) \in P} \max(0, f(x_i, x_k) + \delta - f(x_i, x_j)) + \right. \\ & \left. \sum_{(x_i, x_j, x_0) \in P} \max(0, f(x_i, x_0) + \delta_0 - f(x_i, x_j)) \right] + \\ & \alpha(\mathcal{R}(\theta) + \|X\|_F^2 + \|x_0\|_2^2). \end{aligned} \quad (16)$$

并且建立了两个隐藏层组成的深度模型学习顶点的潜在表示, 如图 7 所示.

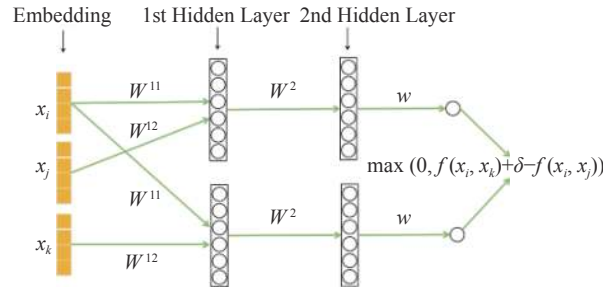


图 7 SiNE 框架<sup>[25]</sup>

Fig. 7 The framework of SiNE

### 3.3.3 GCN

卷积神经网络利用卷积核提取图像里的特征<sup>[26]</sup>, 图像是结构化数据, 网络是非结构化的数据, 图卷积网络 (Graph Convolutional Network, GCN)<sup>[27]</sup> 受谱图理论的启发<sup>[28-29]</sup>, 将卷积的思想发展到图网络上. GCN 通过在图上定义卷积运算, 每一个图卷积层按照非线性函数

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}\right) \quad (17)$$

迭代地聚合顶点邻居的嵌入, 输出新的表示.  $\tilde{A}$  是加了自连接的邻接矩阵,  $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ ,  $W^{(l)}$  是要学习的权重. GCN 的缺点是训练过程慢, 需要训练所有节点才能得到新顶点的表示, 难以应用于大规模网络, 也无法处理动态图任务.

### 3.3.4 GAT

图卷积网络无法处理训练阶段与测试阶段需要处理的网络不同的情况, 有向图的处理也是它的弱点. 注意力机制<sup>[30]</sup> 在循环神经网络 RNN 和卷积神经网络 CNN 中取得了优秀的成效, 图注意力网络 (Graph Attention Network, GAT)<sup>[31]</sup> 在图上引入注意力机制来克服图卷积网络的不足, 对不同的邻居节点分配相应的权重, 重点关注作用较大的节点, 忽视作用较小的节点. GAT 不需要进行矩阵运算, 也不需要事先知道图结构.

所有顶点的输入为  $h = \{\vec{h}_1, \vec{h}_2, \dots, \vec{h}_N\}$ ,  $\vec{h}_i \in \mathbb{R}^F$ , 模型希望得到的输出为  $h' = \{\vec{h}'_1, \vec{h}'_2, \dots, \vec{h}'_N\}$ ,  $\vec{h}'_i \in \mathbb{R}^{F'}$ , 对每个顶点引入注意力机制  $a: \mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R}$ , 得到相关系数:

$$e_{ij} = a(W\vec{h}_i, W\vec{h}_j). \quad (18)$$

$e_{ij}$  表示节点  $j$  对于节点  $i$  的重要性, 其中  $a$  表示前馈神经网络. 再对  $e_{ij}$  进行归一化得到相关系数:

$$\alpha_{ij} = \text{softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})}. \quad (19)$$

在得到归一化系数之后, 就可以预测节点的输出特征表示向量  $\vec{h}'_i = \sigma(\sum_{j \in N_i} \alpha_{ij} W\vec{h}_j)$ . 多头注意力机制的输出通过将所有头的输出进行连接或者平均得到, 如图 8 所示.

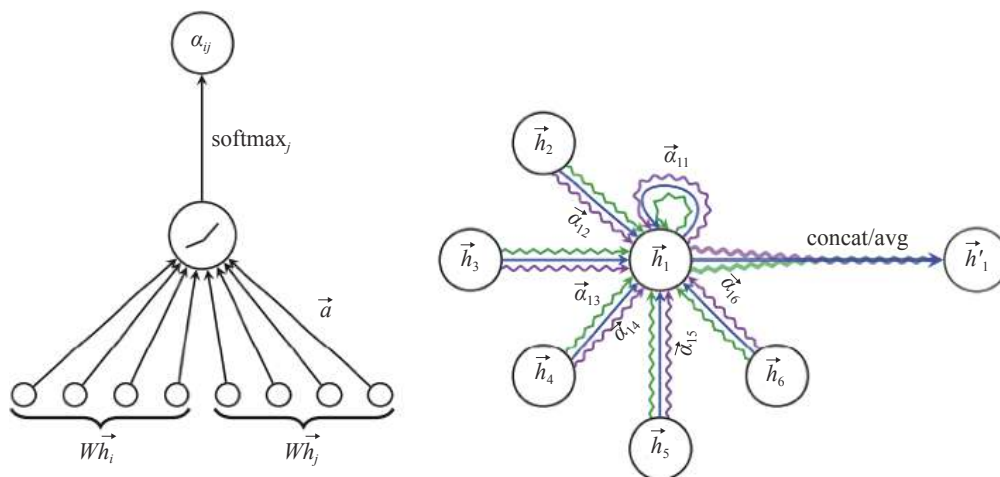


图 8 图注意力机制(左图)和多头注意力(右图)<sup>[31]</sup>

Fig. 8 Illustration of the attention mechanism (left) and multi-head attention (right)

## 4 应用

学习到的网络顶点嵌入可以应用到后续任务中, 如链接预测、顶点分类和可视化. 通常可以根据这些任务上的评价指标来衡量学习到的网络顶点嵌入的优劣.

### 4.1 链接预测

链接预测<sup>[4]</sup>是指预测顶点对之间是否存在边, 这些边可能是缺失的, 也可能在将来网络的不断演化中形成. 在社交网络中, 链接预测技术可以预测两个人是否为朋友关系, 以此推荐可能的好友. 在生物网络中, 链接预测用来判断蛋白质之间是否存在未知的相互作用, 这可以提高在真实世界中实验的目标性, 减少昂贵的实验测试.

### 4.2 顶点分类

顶点分类<sup>[32]</sup>是利用已知标签顶点来推测其他顶点的标签. 在网络中, 顶点会带有标签, 表示实体的属性. 例如, 社交网络中的标签可以表示人的兴趣、喜恶等. 在引用网络中, 标签可以表示文章的关键词或研究领域. 生物网络中顶点的标签可以表示实体的功能. 由于标记成本高或者其他原因, 只有少数顶点被打上了标签, 网络中大部分顶点的标签是未知的.

### 4.3 网络可视化

网络可视化是将原始网络降维到二维空间中以方便展示. 应用主成分分析<sup>[33]</sup>或者 t-SNE<sup>[34]</sup>, 可以

将学习的顶点嵌入进行可视化, 以观察网络的空间结构. 如图 9 所示, a-2) 为使用 DeepWalk 学习到的空手道俱乐部网络 a-1) 的可视化结果, b) 为使用 LINE 学习到的 DBLP 网络的可视化结果. 通过将网络表示进行可视化可以证明模型的有效性.

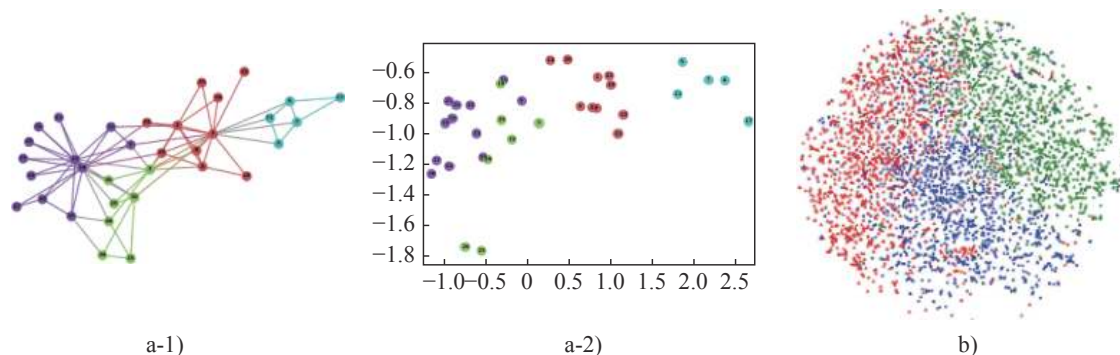


图 9 可视化示例

Fig. 9 Visualization examples

## 5 总 结

本文回顾了近年来网络顶点表示学习的模型和算法, 将现有的模型按照基于矩阵分解、基于随机游走和基于深度学习分为 3 类, 并且介绍了每一类中具有代表性的算法, 算法的目标是能够保留网络的结构和属性特征. 基于矩阵分解的模型适合于小型网络, 模型复杂度较高; 基于随机游走的模型借鉴了自然语言处理领域的思想, 将随机序列视为语句, 顶点视为单词, 再进行后续的分析处理; 新兴的深度学习将自编码器、图卷积技术应用到顶点表示学习中, 适合处理大型的网络. 网络顶点表示学习的未来研究方向是利用非线性模型, 如基于深度学习的算法, 研究不同属性的图, 如符号图、异构图和二分图. 网络嵌入方法还可以应用到知识图谱、生物学网络和社交网络等领域以方便研究.

## [参 考 文 献]

- [1] FREEMAN L C. Visualizing social networks [J]. Social Network Data Analytics, 2000, 6(4): 411-429.
- [2] THEOCHARIDIS A, DONGEN S V, ENRIGHT A J, et al. Network visualization and analysis of gene expression data using BioLayout [J]. Nature Protocol, 2009, 4(10): 1535-1550.
- [3] TANG J, QU M, WANG M, et al. LINE: Large-scale information network embedding [C]// The Web Conference. 2015: 1067-1077.
- [4] LIBENNOWELL D, KLEINBERG J. The link-prediction problem for social networks [J]. Journal of the Association for Information Science and Technology, 2007, 58(7): 1019-1031.
- [5] SEN P, NAMATA G, BILGIC M, et al. Collective classification in network data [J]. Ai Magazine, 2008, 29(3): 93-106.
- [6] HERMAN I, MELANCON G, MARSHALL M S, et al. Graph visualization and navigation in information visualization: A survey [J]. IEEE Transactions on Visualization and Computer Graphics, 2000, 6(1): 24-43.
- [7] ROWEIS S T, SAUL L K. Nonlinear dimensionality reduction by locally linear embedding [J]. Science, 2000, 290(5500): 2323-2326.
- [8] BELKIN M, NIYOGI P. Laplacian eigenmaps and spectral techniques for embedding and clustering [C]// Neural Information Processing Systems. 2001: 585-591.
- [9] CAO S, LU W, XU Q, et al. GraRep: Learning graph representations with global structural information [C]// Conference on Information and Knowledge Management. 2015: 891-900.
- [10] OU M, CUI P, PEI J, et al. Asymmetric transitivity preserving graph embedding [C]// Knowledge Discovery and Data Mining. 2016: 1105-1114.
- [11] MOUSAZADEH S, COHEN I. Embedding and function extension on directed graph [J]. Signal Processing, 2015, 111: 137-149.
- [12] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality [C]// Neural Information Processing Systems. 2013: 3111-3119.
- [13] MIKOLOV T, CHEN K, CORRADO G S, et al. Efficient estimation of word representations in vector space [C]// International Conference on Learning Representations. 2013.
- [14] MIKOLOV T, YIH W, ZWEIG G, et al. Linguistic regularities in continuous space word representations [C]// North American

- Chapter of the Association for Computational Linguistics. 2013: 746-751.
- [15] PEROZZI B, ALRFOU R, SKIENA S, et al. DeepWalk: Online learning of social representations [C]// Knowledge Discovery and Data Mining. 2014: 701-710.
- [16] YANG J, LESKOVEC J. Overlapping communities explain core-periphery organization of networks [J]. Proceedings of the IEEE, 2014, 102(12): 1892-1902.
- [17] GROVER A, LESKOVEC J. Node2vec: Scalable feature learning for networks [C]// Knowledge Discovery and Data Mining. 2016: 855-864.
- [18] NARAYANAN A, CHANDRAMOHAN M, CHEN L, et al. Subgraph2vec: Learning distributed representations of rooted sub-graphs from large graphs [C]// ArXiv: Learning. 2016.
- [19] RIBEIRO L F, SAVERESE P H, FIGUEIREDO D R, et al. Struc2vec: Learning node representations from structural identity [C]// Knowledge Discovery and Data Mining. 2017: 385-394.
- [20] LI C, WANG S, YANG D, et al. PPNE: Property preserving network embedding [C]// Database Systems for Advanced Applications. 2017: 163-179.
- [21] YANG C, LIU Z, ZHAO D, et al. Network representation learning with rich text information [C]// International Conference on Artificial Intelligence. 2015: 2111-2117.
- [22] WANG D, CUI P, ZHU W, et al. Structural deep network embedding [C]// Knowledge Discovery and Data Mining. 2016: 1225-1234.
- [23] CHANG S, HAN W, TANG J, et al. Heterogeneous network embedding via deep architectures [C]// Knowledge Discovery and Data Mining. 2015: 119-128.
- [24] TANG J, CHANG S, AGGARWAL C C, et al. Negative link prediction in social media [C]// Web Search and Data Mining. 2015: 87-96.
- [25] WANG S, TANG J, AGGARWAL C C, et al. Signed network embedding in social media [C]// Siam International Conference on Data Mining. 2017: 327-335.
- [26] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition [C]// Computer Vision and Pattern Recognition. 2016: 770-778.
- [27] KIPF T, WELLING M. Semi-supervised classification with graph convolutional networks [C]// ArXiv: Learning. 2016.
- [28] DEFFERRARD M, BRESSON X, VANDERGHEYNST P, et al. Convolutional neural networks on graphs with fast localized spectral filtering [C]// Neural Information Processing Systems. 2016: 3844-3852.
- [29] NIEPERT M, AHMED M H, KUTZKOV K, et al. Learning convolutional neural networks for graphs [C]// International Conference on Machine Learning. 2016: 2014-2023.
- [30] BAHDANAU D, CHO K, BENGIO Y, et al. Neural machine translation by jointly learning to align and translate [C]// ArXiv: Computation and Language. 2014.
- [31] VELICKOVIC P, CUCURULL G, CASANOVA A, et al. Graph attention networks [C]// ArXiv: Machine Learning. 2017.
- [32] BHAGAT S, CORMODE G, ROZENBAUM I, et al. Applying link-based classification to label blogs [C]// Web Mining and Web Usage Analysis. 2009: 97-117.
- [33] PEARSON K. LIII. On lines and planes of closest fit to systems of points in space [J]. Philosophical Magazine Series 1, 1901, 2(11): 559-572.
- [34] DER MAATEN L V, HINTON G E. Visualizing data using t-SNE [J]. Journal of Machine Learning Research, 2008, 9(11): 2579-2605.

(责任编辑: 李万会)