

# 《DeepWalk: Online Learning of Social Representations》

## 摘要:

1. **论文背景:** 在 NLP 领域, 研究者们已经取得了一系列进展 (如 Word2Vec), 而对于图数据结构, 大多研究者仍在采用人工构造特征的方式来处理问题。
2. **论文的贡献:** 提出了一种新的无监督学习算法 **DeepWalk** 用于图表示学习, 并且是**第一个**将 NLP 中的思想推广到网络嵌入 (Network Embedding) 上的工作, 为图神经网络的发展迈出了坚实的一步。
3. **主要创新点:** 通过采用随机游走的方式来获取一系列图的结构序列, 然后利用 NLP 中 Word2Vec 方法里的 Skip-Gram 方式来无监督地学习图结构的嵌入表示。更进一步地说, 本论文使得图问题与自然语言处理问题在研究形式上初步实现了统一。
4. **实验分析:** 实验表明了 DeepWalk 的性能远优于**所有**传统的图节点分类算法, 尤其是在存在缺失信息的情况下。即, 当标记的数据稀疏时, 利用 DeepWalk 的嵌入表示可以提供比其他方法高出 10% 的 F1 分数。此外, DeepWalk 还有一种可扩展的在线学习算法, 可构建有用的增量结果, 并且易于并行化, 而上述这些特性也使其能够适用于广泛的实际应用。**【现如今已被广泛应用, 且它的影响深远。】**

## 问题定义:

现假设图 $G$ 的定义如下, 它由顶点集 $V$ 和边集 $E$ 组成:

$$G = (V, E), \quad E \subseteq (V, V)$$

而如果在图 $G$ 的基础上再加上节点的向量表示和节点所属的标签 (在网络节点分类问题中, 网络中的每个节点都有一个类别, 而所属的类别即为该节点的标签 label) 就构成了一个标签图(labeled graph), 即:

$$G_L = (V, E, X, Y), \quad E \subseteq (V, V), \quad X \in \mathbb{R}^{|V| \times S}, \quad Y \in \mathbb{R}^{|V| \times |Y|}$$

其中,  $S$ 表示节点的特征 (向量维数),  $Y$ 表示对应的标签集。

然而, 由于**图连接的稀疏性**, 即边集 $E$ 所构成的矩阵存在大部分 0 填充, 使得很难将上述图问题用机器学习的方式进行解决。而本文 DeepWalk 的提出正是有效地解决了这种稀疏性问题。简单来说, DeepWalk 不再像之前的传统方法

一样试图将标签和连接特征混合，而是尝试在 embedding 中仅编码节点的连接信息。【DeepWalk 是无监督学习，它与节点 label 和节点本身的特征无关。】

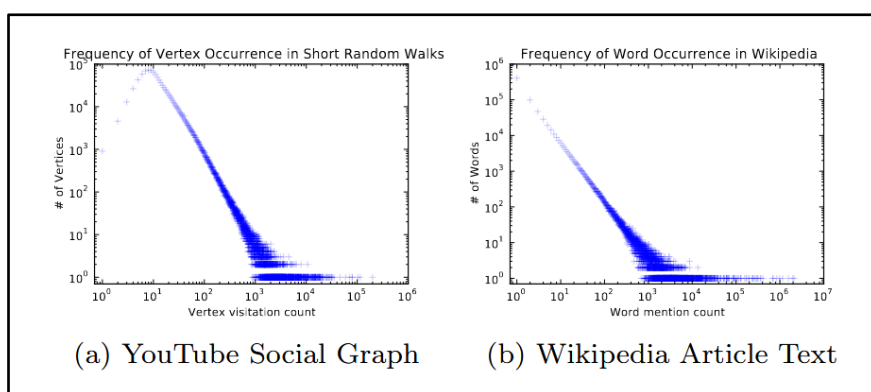
### 图网络嵌入：

在学习一个图网络表示时，我们需要注意以下几个性质：

- (1) **适应性**：网络表示必须能够适应网络的变化。由于网络是一个动态的图，会不断地有新的节点和边添加进来，所以网络表示需要适应网络的正常演化。
- (2) **能够反映社群聚类信息**：网络中往往会出现一些特征相似的节点构成的团状结构，而这些节点在表示成向量后必须相似。
- (3) **低维度**：代表每个节点的嵌入向量维数不能过高，因为过高会有过拟合的风险。
- (4) **连续性**：低维向量表示必须是连续的。

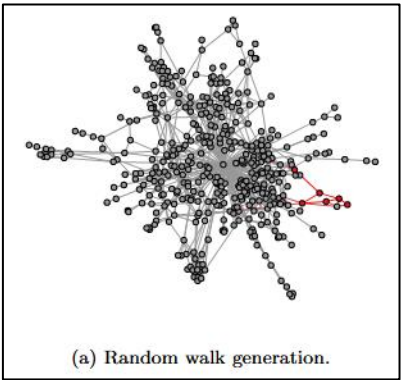
而本文处理图节点的表示 (node representation) 则是利用了词嵌入 Word2Vec (词向量) 的思想。两者的对比如下：词嵌入的基本处理元素是单词，而网络节点嵌入的基本处理元素是网络节点；词嵌入是对构成一个句子中的单词序列进行分析，而网络节点嵌入是对节点构成的序列 (随机游走) 进行分析。

### 幂律分布：



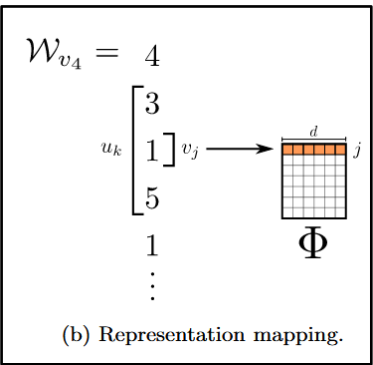
如上图所示，网络中随机游走的分布规律与 NLP 中句子序列在语料库中出现的规律有着类似的幂律分布特征。因此，这也侧面印证了网络的特性实际上与自然语言处理中的特性十分相似。所以，就更可以将 NLP 中词向量的模型用在网络表示中，而这正是本文所做的工作。

随机游走生成:



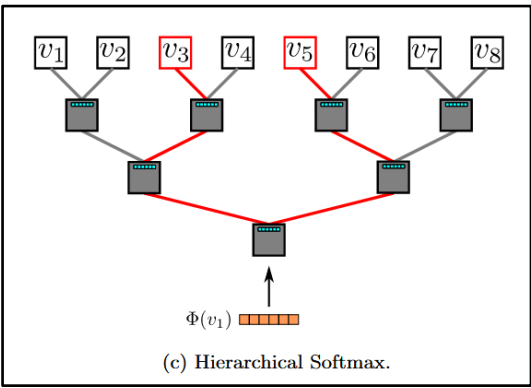
如图所示，我们在图网络中选择一给定节点 $v_i$ ，然后以 $v_i$ 为起始点随机连接到与 $v_i$ 关联的下一个节点（邻居）上，并如此重复  $k$  次，从而形成随机游走序列  $W_{v_i}^1, W_{v_i}^2, W_{v_i}^3, \dots, W_{v_i}^k$ ，其中 $W_{v_i}^k$ 表示以 $v_i$ 为起点的第  $k$  次游走。

映射表示:



如图所示，每个节点的  $d$  维嵌入向量代表嵌入矩阵 $\Phi$ 中的对应一行。

分级 softmax:



一种优化（加速）方法。【对应的另一种优化方法是：负采样】

DeepWalk 算法:

简单来说, DeepWalk 算法主要包括两个步骤——第一步: 随机游走采样节点序列; 第二步: 使用 Word2Vec Model 中的 Skip-Gram 方式学习表达向量。

Algorithm 1 DEEPWALK( $G, w, d, \gamma, t$ )

Input: graph  $G(V, E)$   
window size  $w$   
embedding size  $d$   
walks per vertex  $\gamma$   
walk length  $t$

Output: matrix of vertex representations  $\Phi \in \mathbb{R}^{|V| \times d}$

1: Initialization: Sample  $\Phi$  from  $\mathcal{U}^{|V| \times d}$

2: Build a binary Tree  $T$  from  $V$

3: for  $i = 0$  to  $\gamma$  do

4:    $\mathcal{O} = \text{Shuffle}(V)$

5:   for each  $v_i \in \mathcal{O}$  do

6:      $\mathcal{W}_{v_i} = \text{RandomWalk}(G, v_i, t)$

7:     SkipGram( $\Phi, \mathcal{W}_{v_i}, w$ )

8:   end for

9: end for

其中,  $w$ 代表左右窗口宽度,  $d$ 代表 embedding 的维度,  $\gamma$ 代表每个节点作为起始节点生成随机游走的次数,  $t$ 代表每次随机游走的最大长度。算法的目标是获得一个节点表示矩阵 $\Phi$ , 而在一开始我们会随机初始化 $\Phi$ 。

Algorithm 2 SkipGram( $\Phi, \mathcal{W}_{v_i}, w$ )

1: for each  $v_j \in \mathcal{W}_{v_i}$  do

2:   for each  $u_k \in \mathcal{W}_{v_i}[j - w : j + w]$  do

3:      $J(\Phi) = -\log \Pr(u_k | \Phi(v_j))$

4:      $\Phi = \Phi - \alpha * \frac{\partial J}{\partial \Phi}$

5:   end for

6: end for

而在进一步的 Skip-Gram 算法中详细阐述了如何利用梯度下降法来对 $\Phi$ 进行优化。

实验结果:

	% Labeled Nodes	10%	20%	30%	40%	50%	60%	70%	80%	90%
Micro-F1(%)	DEEPWALK	36.00	38.20	39.60	40.30	41.00	41.30	41.50	41.50	42.00
	SpectralClustering	31.06	34.95	37.27	38.93	39.97	40.99	41.66	42.42	42.62
	EdgeCluster	27.94	30.76	31.85	32.99	34.12	35.00	34.63	35.99	36.29
	Modularity	27.35	30.74	31.77	32.97	34.09	36.13	36.08	37.23	38.18
	wvRN	19.51	24.34	25.62	28.82	30.37	31.81	32.19	33.33	34.28
	Majority	16.51	16.66	16.61	16.70	16.91	16.99	16.92	16.49	17.26
Macro-F1(%)	DEEPWALK	21.30	23.80	25.30	26.30	27.30	27.60	27.90	28.20	28.90
	SpectralClustering	19.14	23.57	25.97	27.46	28.31	29.46	30.13	31.38	31.78
	EdgeCluster	16.16	19.16	20.48	22.00	23.00	23.64	23.82	24.61	24.92
	Modularity	17.36	20.00	20.80	21.85	22.65	23.41	23.89	24.20	24.97
	wvRN	6.25	10.13	11.64	14.24	15.86	17.18	17.98	18.86	19.57
	Majority	2.52	2.55	2.52	2.58	2.58	2.63	2.61	2.48	2.62

Table 2: Multi-label classification results in BLOGCATALOG

	% Labeled Nodes	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1(%)	DEEPWALK	<b>32.4</b>	<b>34.6</b>	<b>35.9</b>	<b>36.7</b>	<b>37.2</b>	<b>37.7</b>	<b>38.1</b>	<b>38.3</b>	<b>38.5</b>	<b>38.7</b>
	SpectralClustering	27.43	30.11	31.63	32.69	33.31	33.95	34.46	34.81	35.14	35.41
	EdgeCluster	25.75	28.53	29.14	30.31	30.85	31.53	31.75	31.76	32.19	32.84
	Modularity	22.75	25.29	27.3	27.6	28.05	29.33	29.43	28.89	29.17	29.2
	wvRN	17.7	14.43	15.72	20.97	19.83	19.42	19.22	21.25	22.51	22.73
	Majority	16.34	16.31	16.34	16.46	16.65	16.44	16.38	16.62	16.67	16.71
Macro-F1(%)	DEEPWALK	<b>14.0</b>	<b>17.3</b>	<b>19.6</b>	<b>21.1</b>	<b>22.1</b>	<b>22.9</b>	<b>23.6</b>	<b>24.1</b>	<b>24.6</b>	<b>25.0</b>
	SpectralClustering	13.84	<b>17.49</b>	19.44	20.75	21.60	22.36	23.01	23.36	23.82	24.05
	EdgeCluster	10.52	14.10	15.91	16.72	18.01	18.54	19.54	20.18	20.78	20.85
	Modularity	10.21	13.37	15.24	15.11	16.14	16.64	17.02	17.1	17.14	17.12
	wvRN	1.53	2.46	2.91	3.47	4.95	5.56	5.82	6.59	8.00	7.26
	Majority	0.45	0.44	0.45	0.46	0.47	0.44	0.45	0.47	0.47	0.47

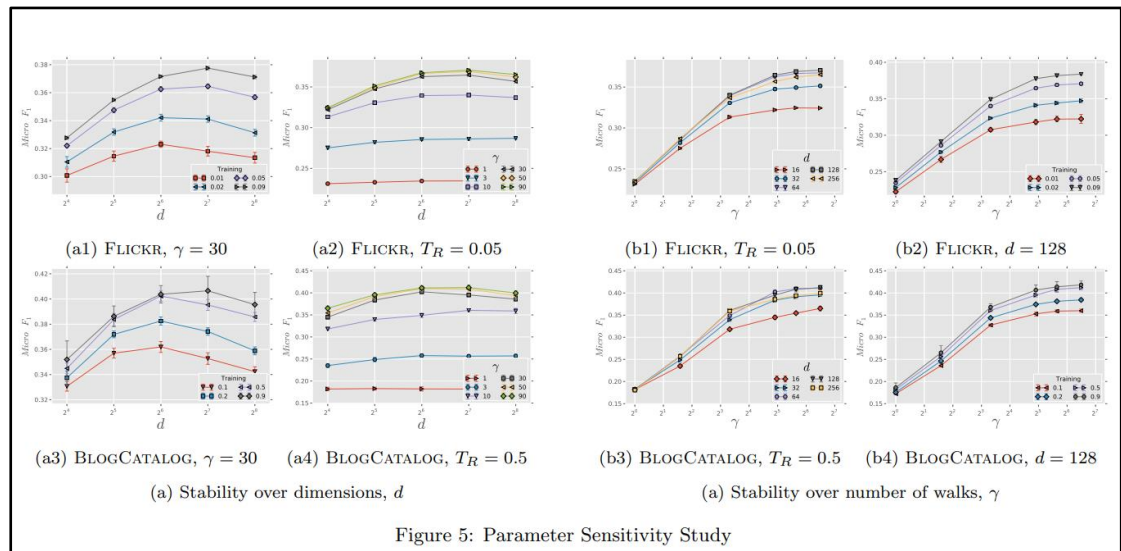
Table 3: Multi-label classification results in FLICKR

	% Labeled Nodes	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1(%)	DEEPWALK	<b>37.95</b>	<b>39.28</b>	<b>40.08</b>	<b>40.78</b>	<b>41.32</b>	<b>41.72</b>	<b>42.12</b>	<b>42.48</b>	<b>42.78</b>	<b>43.05</b>
	SpectralClustering	—	—	—	—	—	—	—	—	—	—
	EdgeCluster	23.90	31.68	35.53	36.76	37.81	38.63	38.94	39.46	39.92	40.07
	Modularity	—	—	—	—	—	—	—	—	—	—
	wvRN	26.79	29.18	33.1	32.88	35.76	37.38	38.21	37.75	38.68	39.42
	Majority	24.90	24.84	25.25	25.23	25.22	25.33	25.31	25.34	25.38	25.38
Macro-F1(%)	DEEPWALK	<b>29.22</b>	<b>31.83</b>	<b>33.06</b>	<b>33.90</b>	<b>34.35</b>	<b>34.66</b>	<b>34.96</b>	<b>35.22</b>	<b>35.42</b>	<b>35.67</b>
	SpectralClustering	—	—	—	—	—	—	—	—	—	—
	EdgeCluster	19.48	25.01	28.15	29.17	29.82	30.65	30.75	31.23	31.45	31.54
	Modularity	—	—	—	—	—	—	—	—	—	—
	wvRN	13.15	15.78	19.66	20.9	23.31	25.43	27.08	26.48	28.33	28.89
	Majority	6.12	5.86	6.21	6.1	6.07	6.19	6.17	6.16	6.18	6.19

Table 4: Multi-label classification results in YOUTUBE

如图所示，DeepWalk 在各项任务中都显著优于其他的传统算法。

### 参数灵敏度研究：



如图所示，我们可以发现四个结论：1)  $T_R$ (训练比率)会影响最优的  $d$ ；2)  $\gamma$  越大，效果越好，但边际效果逐渐降低；3)  $T_R$  越大，效果越好；4) 不同大小的 graph，不同  $\gamma$  的相对影响是一致的。

## 总结：

本文提出了一种学习图节点潜在表示的新方法——DeepWalk。它通过使用随机游走生成的局部信息序列作为输入，来学习和编码图结构的嵌入表示。并且，在各种不同的图上的实验研究也有力证明了该方法的有效性。

## 对本文的感悟：

DeepWalk 将词嵌入思想有效推广到了图嵌入领域，为图表示学习、图神经网络提供了一种全新的思路。并且，它还潜在揭示了图和自然语言之间密不可分的联系（“可见 graph 与不可见 graph”）。总之，DeepWalk 为之后图领域的研究提供了一条切实可行的道路。