DeepWalk: Online Learning of Social Representations

Perozzi B, Al-Rfou R, Skiena S

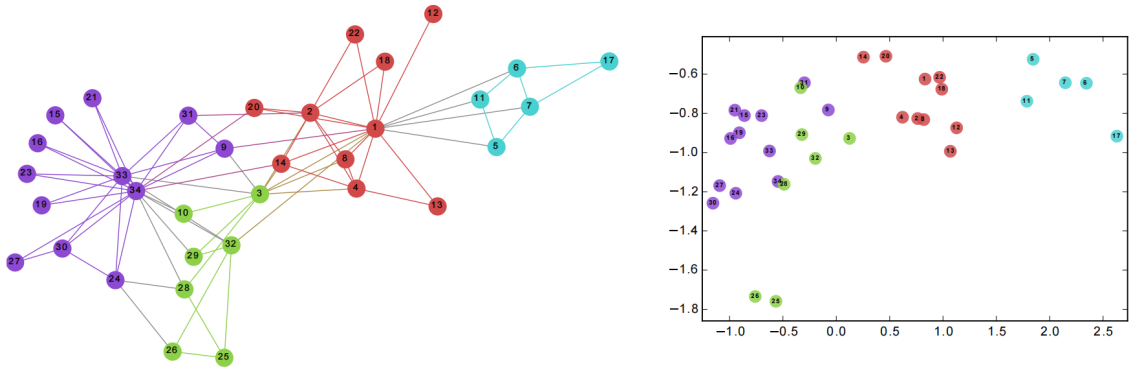
KDD 2014

报告人：石智超

2022 年 12 月 5 日

Introduction

网络嵌入表示即将网络节点用一个低维连续空间中的向量表示，且这些向量能够反映原先网络的结构、属性等特性。



(a) Input: Karate Graph (b) Output: Representation

论文提出 DeepWalk，通过建模网络上**短的随机游走**，利用**语言模型**处理随机游走构成的“语句”生成表征网络结构、关系等特性的节点潜在嵌入向量表示。

Learning Social Representations

论文认为，网络节点向量表征需要具备如下特征：

- **适应性**：适应网络动态演化而不需要反复重新学习
- **社区相似**：向量距离应能衡量节点相似度，且适用于同质网络
- **低维**：向量表示维数不能过高
- **连续性**：潜在向量空间连续以建立更平滑稳健的决策分类

提及嵌入表示，能够自然联想到 word2vec：

嵌入表示	基本处理元素	分析序列
word2vec	单词	句子中单词序列
网络嵌入	节点	随机游走中节点序列

Learning Social Representations

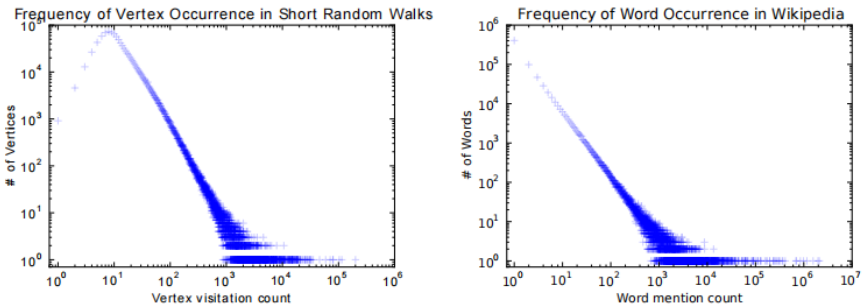
随机游走：

使用随机游走作为从网络提取信息的基本工具，具有两大特性：

- 对网络局部信息的探取过程易于并行化
- 有效避免在网络结构发生微小变化时全局进行重新计算

幂律分布：

在节点度具有幂律分布的连通图（例如无标度网络）中，短随机游走中节点出现频率同样遵循幂律分布；同时，自然语言中单词频率也服从幂律分布。既然**网络中节点分布特性与 NLP 中单词分布特性十分类似**，那么就可以将 NLP 中词向量的模型用在网络嵌入表示中。



(a) YouTube Social Graph (b) Wikipedia Article Text

Figure 2: The distribution of vertices appearing in short random walks (2a) follows a power-law, much like the distribution of words in natural language (2b).

Learning Social Representations

语言模型：

对于给定的单词序列 $W_1^n = (w_0, w_1, \ldots, w_n)$，词向量模型优化目标是最大化 $\Pr(w_n|(w_0, w_1, \cdots, w_{n-1}))$。对应到网络上，就是给定随机游走 $(v_0, v_1, \ldots, v_n)$，优化目标是最大化 $\Pr(v_n|(v_0, v_1, \ldots, v_{n-1}))$。

模型目标在于学习节点潜在向量表示，而不只是共现概率分布，因此引入节点到向量空间的映射函数 $\Phi: v \in V \rightarrow \mathbb{R}^{|V| \times d}$，优化目标转换为：

$$\text{maximize } \Pr(v_n|\big(\Phi(v_0), \Phi(v_1), \ldots, \Phi(v_{n-1})\big))$$

但随着游走长度增长，目标函数的计算变得十分困难。语言模型中进行了松弛处理：1）改为使用一个单词预测上下文；2）上下文包含目标两侧单词；3）不考虑单词顺序。在网络上应用该模型，优化目标变为：

$$\underset{\Phi}{\text{minimize}} \ -\log \Pr\Big(\{v_{i-w}, \ldots, v_{i+w}\} \backslash v_i | \Phi(v_i)\Big)$$

Method: Algorithm

Algorithm 1 DeepWalk(G, w, d, γ, t)

Input: graph $G(V, E)$
window size w
embedding size d
walks per vertex γ
walk length t
Output: matrix of vertex representations $\Phi \in \mathbb{R}^{|V| \times d}$

Initialization: Sample Φ from $\mathcal{U}^{|V| \times d}$
Build a binary Tree T from V
for $i = 0$ to γ **do**
  $\mathcal{O} = \text{Shuffle}(V)$
  for each $v_i \in \mathcal{O}$ **do**
    $\mathcal{W}_{v_i} = RandomWalk(G, v_i, \text{t})$
    SkipGram(Φ, $\mathcal{W}_{v_i}$, w)
  end for
end for

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

for each $v_j \in \mathcal{W}_{v_i}$ **do**
  for each $u_k \in \mathcal{W}_{v_i}[j - w : j + w]$ **do**
    $J(\Phi) = -\log \Pr(u_k \mid \Phi(v_j))$
    $\Phi = \Phi - \alpha * \frac{\partial J}{\partial \Phi}$
  end for
end for

Method: Optimization

给定节点 v_j，计算窗口 w 中概率 $\mathrm{Pr}(u_k|\Phi(v_j))$ 代价昂贵，所以使用分层 softmax 进行优化。构建二叉树，叶子节点为图节点，其余为中间节点，每个中间节点维护一个向量，训练过程中不断更新。

$\mathrm{Pr}(u_k|\Phi(v_j))$ 就可表示为路径 $(b_0, b_1, \ldots, b_{\lceil \log |V| \rceil} = u_k)$ 上的概率累积：

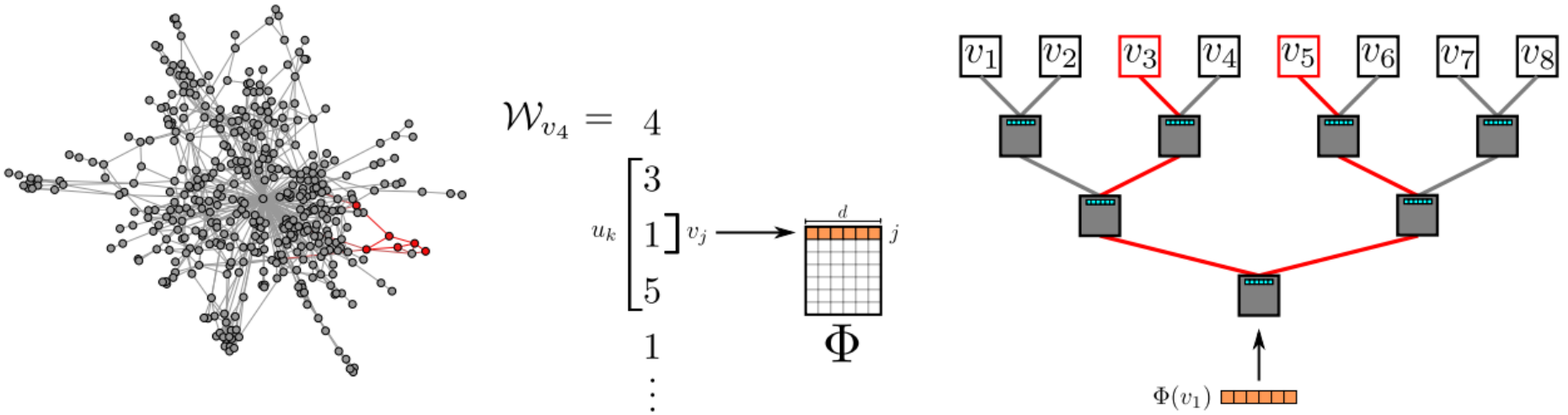
$$\mathrm{Pr}(u_k|\Phi(v_j)) = \prod_{l=1}^{\lceil \log |V| \rceil} \mathrm{Pr}(b_l|\Phi(v_j))$$

对于中间节点，概率计算如下：

$$\mathrm{Pr}(b_l|\Phi(v_j)) = \begin{cases} 1/(1+e^{-\Phi(v_j)\Psi(b_l)}), & left \\ 1-1/(1+e^{-\Phi(v_j)\Psi(b_l)}), & right \end{cases}$$

$\Psi(b_l)$ 是树节点 b_l 父节点维护的向量

Method: Overview



(a) Random walk generation. (b) Representation mapping. (c) Hierarchical Softmax.

Figure 3: Overview of DeepWalk. We slide a window of length $2w+1$ over the random walk $\mathcal{W}_{v_4}$, mapping the central vertex v_1 to its representation $\Phi(v_1)$. Hierarchical Softmax factors out $\Pr(v_3 \mid \Phi(v_1))$ and $\Pr(v_5 \mid \Phi(v_1))$ over sequences of probability distributions corresponding to the paths starting at the root and ending at v_3 and v_5. The representation Φ is updated to maximize the probability of v_1 co-occurring with its context $\{v_3, v_5\}$.

Experiments

Datasets: BlogCatalog, Flicker, YouTube

Name	BlogCatalog	Flickr	YouTube
$\|V\|$	10,312	80,513	1,138,499
$\|E\|$	333,983	5,899,882	2,990,443
$\|\mathcal{Y}\|$	39	195	47
Labels	Interests	Groups	Groups

Table 1: Graphs used in our experiments.

Baselines:

- **SpectralClustering:** 谱聚类，通过归一化拉普拉斯矩阵 $\tilde{L}$ 最小 d 个特征向量生成嵌入表示
- **Modularity:** 通过模块度矩阵 B 的 $top-d$ 特征向量生成嵌入表示
- **EdgeCluster:** 使用 k-means 聚类 G 的邻接矩阵
- **wvRN:** 通过邻节点概率加权和计算节点从属类别 y_i 的概率

$$\Pr(y_i|\mathcal{N}_i) = \frac{1}{Z}\sum_{v_j \in \mathcal{N}_i} w_{ij}\Pr(y_i|\mathcal{N}_j)$$

- **Majority:** 粗暴地选择训练集频率最高的类别作为预测类别

Experiments: Multi-Label Classification Results

	% 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	**32.4**	**34.6**	**35.9**	**36.7**	**37.2**	**37.7**	**38.1**	**38.3**	**38.5**	**38.7**
	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	**14.0**	17.3	**19.6**	**21.1**	**22.1**	**22.9**	**23.6**	**24.1**	**24.6**	**25.0**
	SpectralClustering	13.84	**17.49**	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

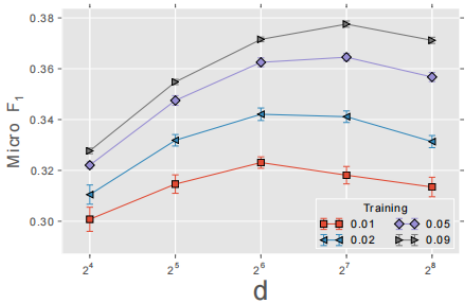
Table 3: Multi-label classification results in Flickr

Experiments: Multi-Label Classification Results

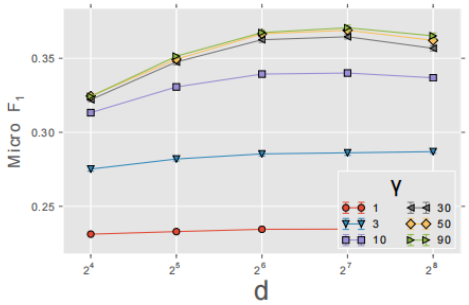
	% Labeled Nodes	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%
Micro-F1(%)	DeepWalk	**37.95**	**39.28**	**40.08**	**40.78**	**41.32**	**41.72**	**42.12**	**42.48**	**42.78**	**43.05**
	SpectralClustering	—	—	—	—	—	—	—	—	—	—
	EdgeCluster	23.90	31.68	35.53	36.76	37.81	38.63	38.94	39.46	39.92	40.07
	Modularity	—	—	—	—	—	—	—	—	—	—
	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	**29.22**	**31.83**	**33.06**	**33.90**	**34.35**	**34.66**	**34.96**	**35.22**	**35.42**	**35.67**
	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

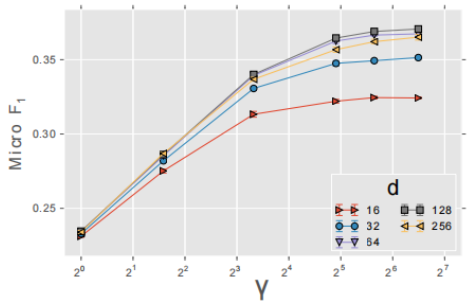
Experiments: Parameter Sensitivity Study



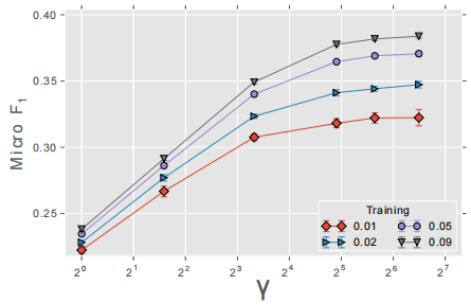
(a1) FLICKR, $\gamma = 30$



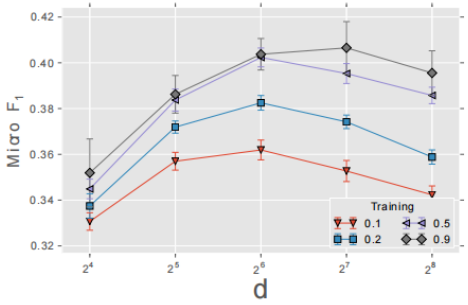
(a2) FLICKR, $T_R = 0.05$



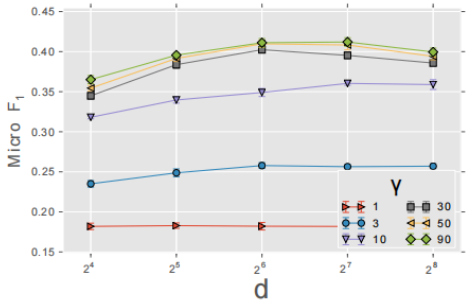
(b1) FLICKR, $T_R = 0.05$



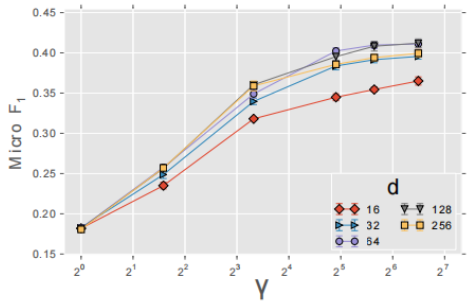
(b2) FLICKR, $d = 128$



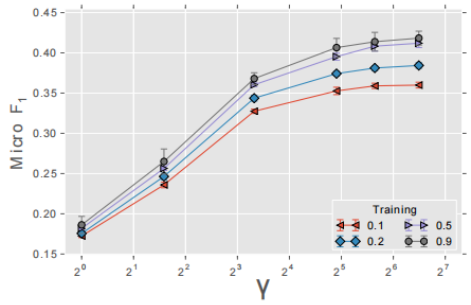
(a3) BLOGCATALOG, $\gamma = 30$



(a4) BLOGCATALOG, $T_R = 0.5$



(b3) BLOGCATALOG, $T_R = 0.5$



(b4) BLOGCATALOG, $d = 128$

(a) Stability over dimensions, d

(b) Stability over number of walks, γ

Thanks