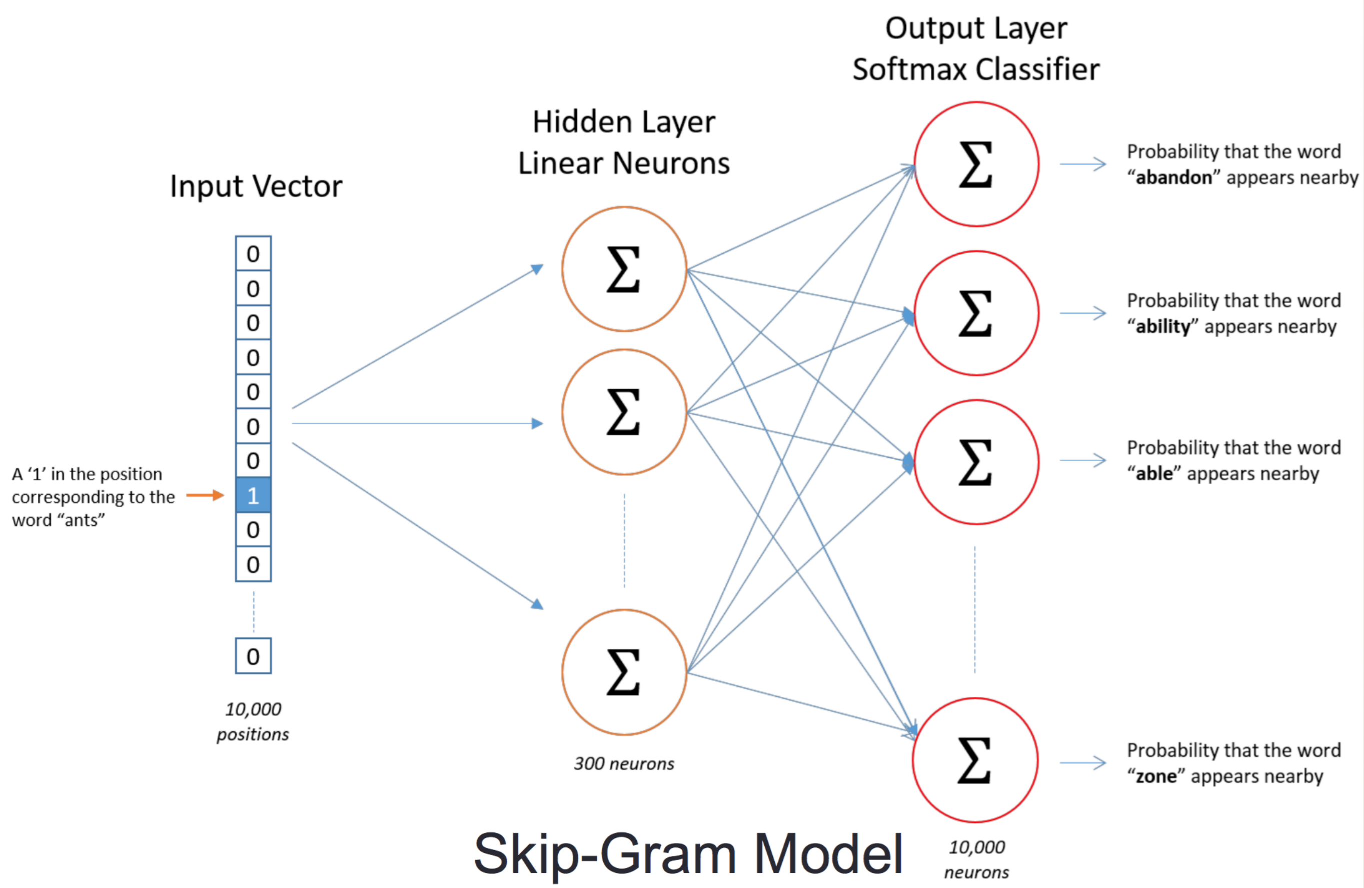
NODE2VEC

KDD 2016

宋军帅

2017/03/22

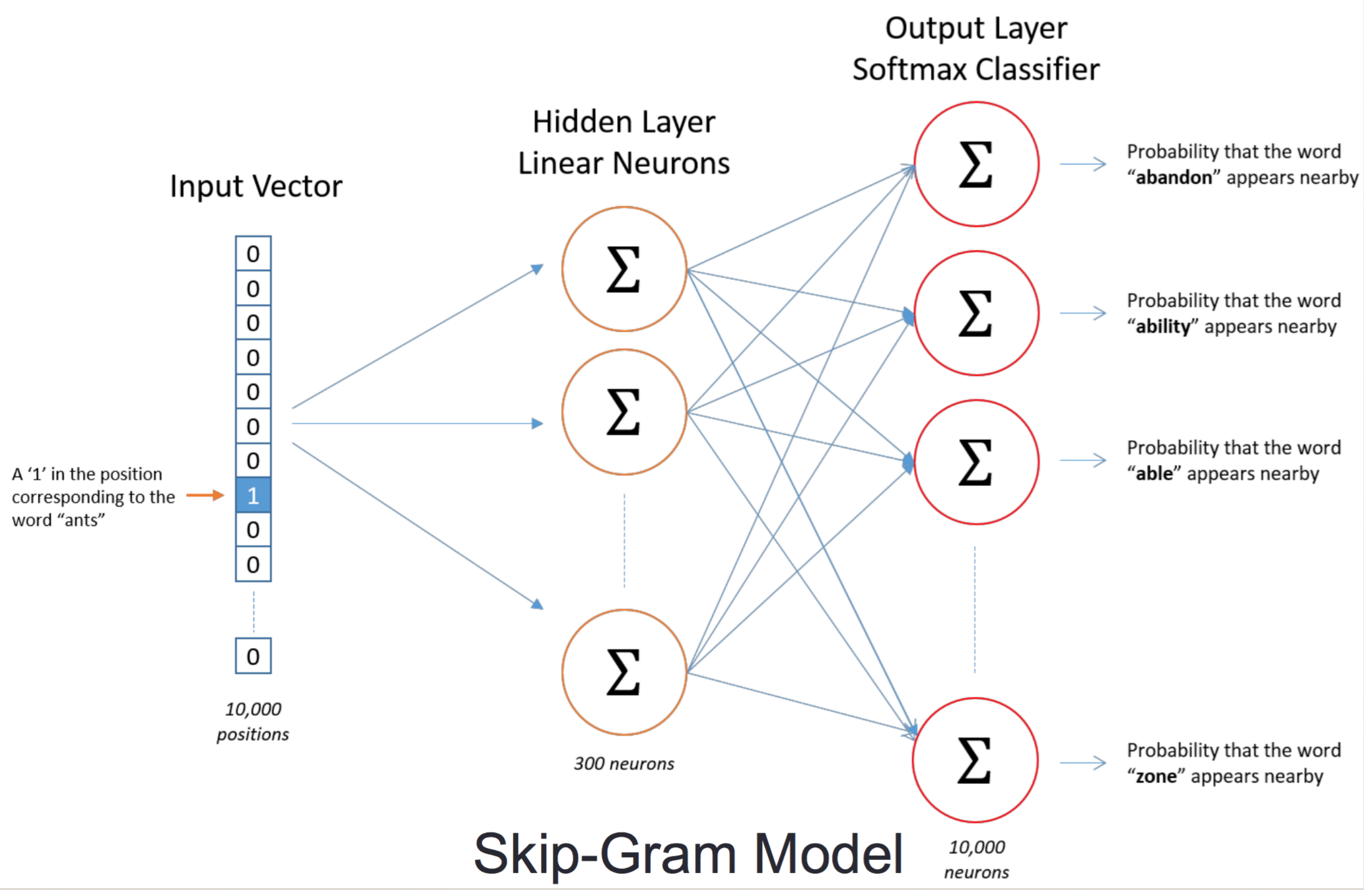
SKIP-MODEL



- This is a pen.
- Input: [0,0,…0,0,1,0,0]
- Output: [0,0,…0,1,0,1,0]

- This is a pen.
- Input: [0,0,…0,0,0,1,0]
- Output: [0,0,…0,0,1,0,1]

SKIP-MODEL



- 流程：
- 输入one-hot向量经过一个隐层神经网络
- 输出层经过一个SoftMax得概率分布y
- 真实的概率分布为Y
- 我们可以得到单个样本交叉熵损失：

$$E(i) = -\sum_{j=0}^{10000} Y_j \log y_j$$

PRELIMINARY – DEEPWALK

- 第一次将word2vec模型应用在了Graph上，
- 单词对应Graph上顶点，句子表示从Graph上某顶点随机采样的一条路径。
- 在DeepWalk中，从各个顶点，每个顶点采样一定数量路径作为所有的句子，之后使用word2vec来建模所有顶点特征表达。

PRELIMINARY – DEEPWALK

Random Walks on Graph

- $V_{26} - V_{25} - V_{32} - V_{3} - V_{10} \ldots$
- $V_{5} - V_{7} - V_{17} - V_{6} - V_{11} \ldots$
- $V_{31} - V_{33} - V_{21} - V_{33} - V_{15}$

NODE2VEC

- Introduction
- Related Work
  - Word2vec/DeepWalk
- Feature Learning Framework
  - Maximizes the log-probability
  - Classic search strategies
  - Node2vec
  - Learning edge features
- Experiments

INTRODUCTION

- DeepWalk
- Present feature learning approaches are not expressive enough to capture the diversity of connectivity patterns observed in networks.
  - --- rigid search procedures in prior work.

- In Node2vec,The added flexibility in exploring neighborhoods is the key to learning richer representations.
  - --- Use a 2nd order random walk approach to generate (sample) network neighborhoods for nodes .

- Social Relations
  - We seek to optimize the following objective function, which maximizes the log-probability of observing a network neighborhood NS (u) for a node u conditioned on its feature representation, given by f:

$$\max_f \sum_{u \in V} \log Pr(N_S(u) | f(u)).$$

  - Conditional independence

$$Pr(N_S(u)|f(u)) = \prod_{n_i \in N_S(u)} Pr(n_i|f(u))$$

  - Symmetry in feature space

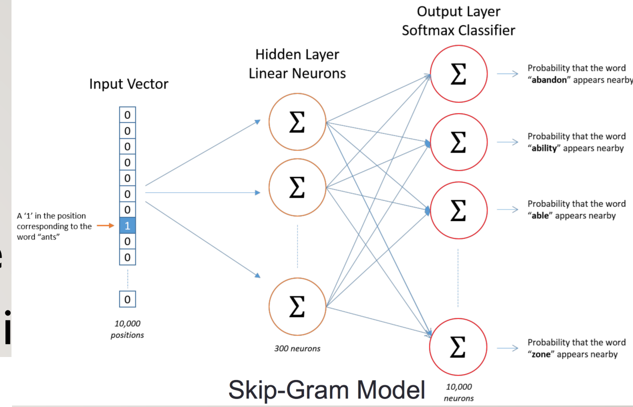
$$Pr(n_i|f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))}$$

  - Objective：

$$\max_f \sum_{u \in V} \Big[-\log Z_u + \sum_{n_i \in N_S(u)} f(n_i) \cdot f(u) \Big]$$

- Social Relations
  - We seek to optimize the following objective function, which maximizes the
    of observing a network neighborhood NS (u) for a node u conditioned on i
    representation, given by f:

$$\max_f \sum_{u \in V} \log Pr(N_S(u) | f(u)).$$



  - Conditional independence

$$Pr(N_S(u)|f(u)) = \prod_{n_i \in N_S(u)} Pr(n_i|f(u))$$

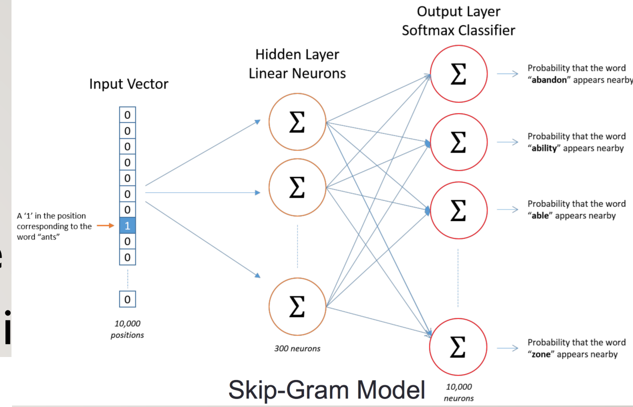
  - Symmetry in feature space

$$Pr(n_i|f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))}$$

  - Objective：

$$\max_f \sum_{u \in V} \Big[-\log Z_u + \sum_{n_i \in N_S(u)} f(n_i) \cdot f(u) \Big]$$

- Social Relations
  - We seek to optimize the following objective function, which maximizes the of observing a network neighborhood NS (u) for a node u conditioned on i representation, given by f:



$$\max_{f} \sum_{u \in V} \log Pr(N_S(u) | f(u)).$$

  - Conditional independence

$$Pr(N_S(u)|f(u)) = \prod_{n_i \in N_S(u)} Pr(n_i|f(u))$$

  - Symmetry in feature space

$$Pr(n_i|f(u)) = \frac{\exp(f(n_i) \cdot f(u))}{\sum_{v \in V} \exp(f(v) \cdot f(u))}$$

  - Objective：

$$\max_{f} \sum_{u \in V} \Big[-\log Z_u + \sum_{n_i \in N_S(u)} f(n_i) \cdot f(u) \Big]$$

  - $Ns(u)$ ？

PRELIMINARY – DEEPWALK

Random Walks on Graph

- $V_{26} - V_{25} - V_{32} - V_{3} - V_{10} \ldots$
- $V_{5} - V_{7} - V_{17} - V_{6} - V_{11} \ldots$
- $V_{31} - V_{33} - V_{21} - V_{33} - V_{15}$

- Node2vec

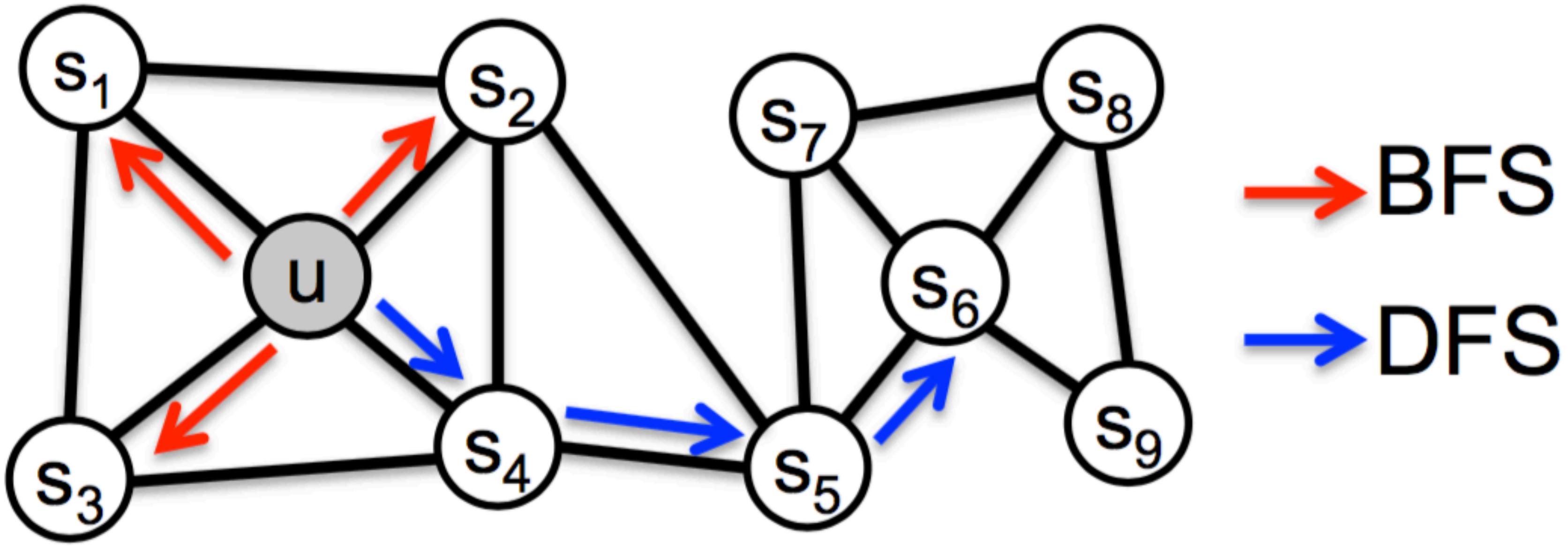
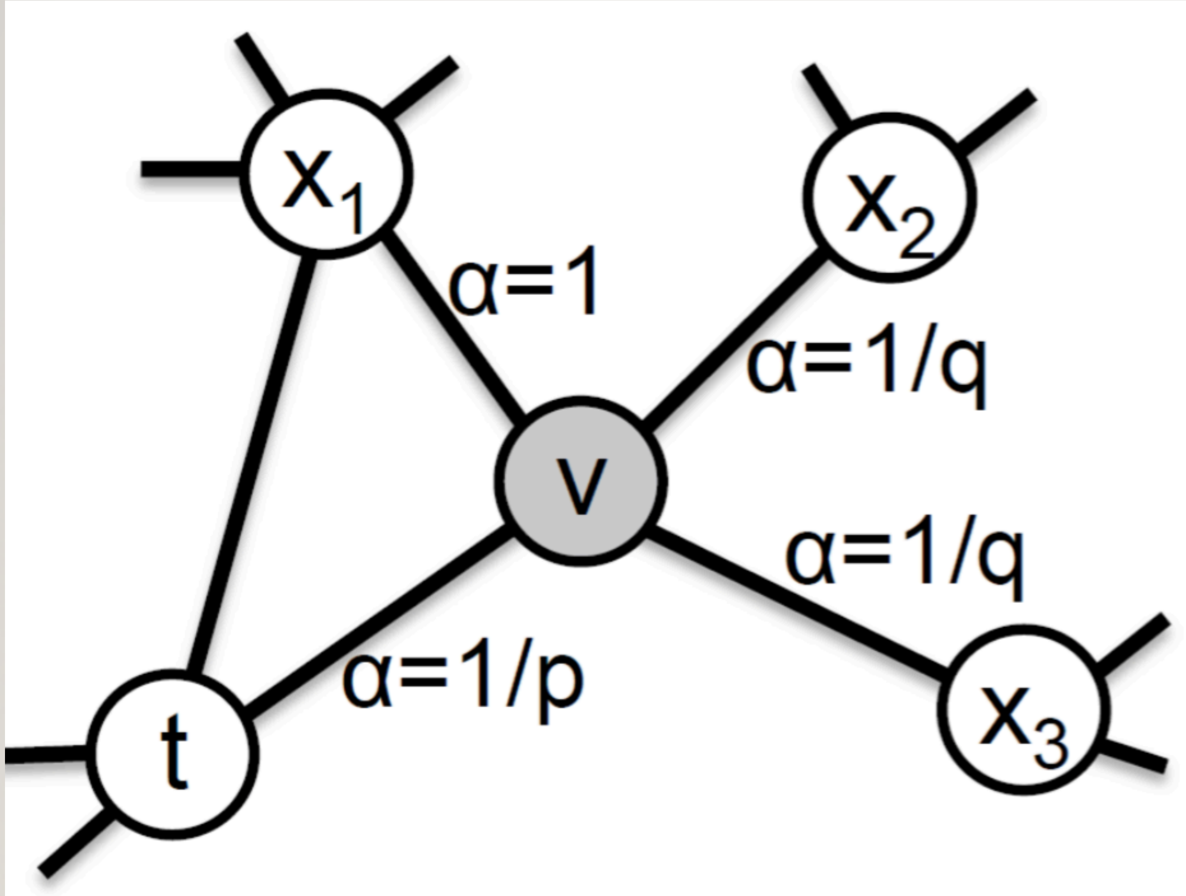


Figure 1: BFS and DFS search strategies from node u ($k = 3$).

HOW TO PRESERVE BOTH RELATIONS USING A UNIFIED METHOD?

It's key contribution is in defining a flexible notion of a node's network neighborhood.

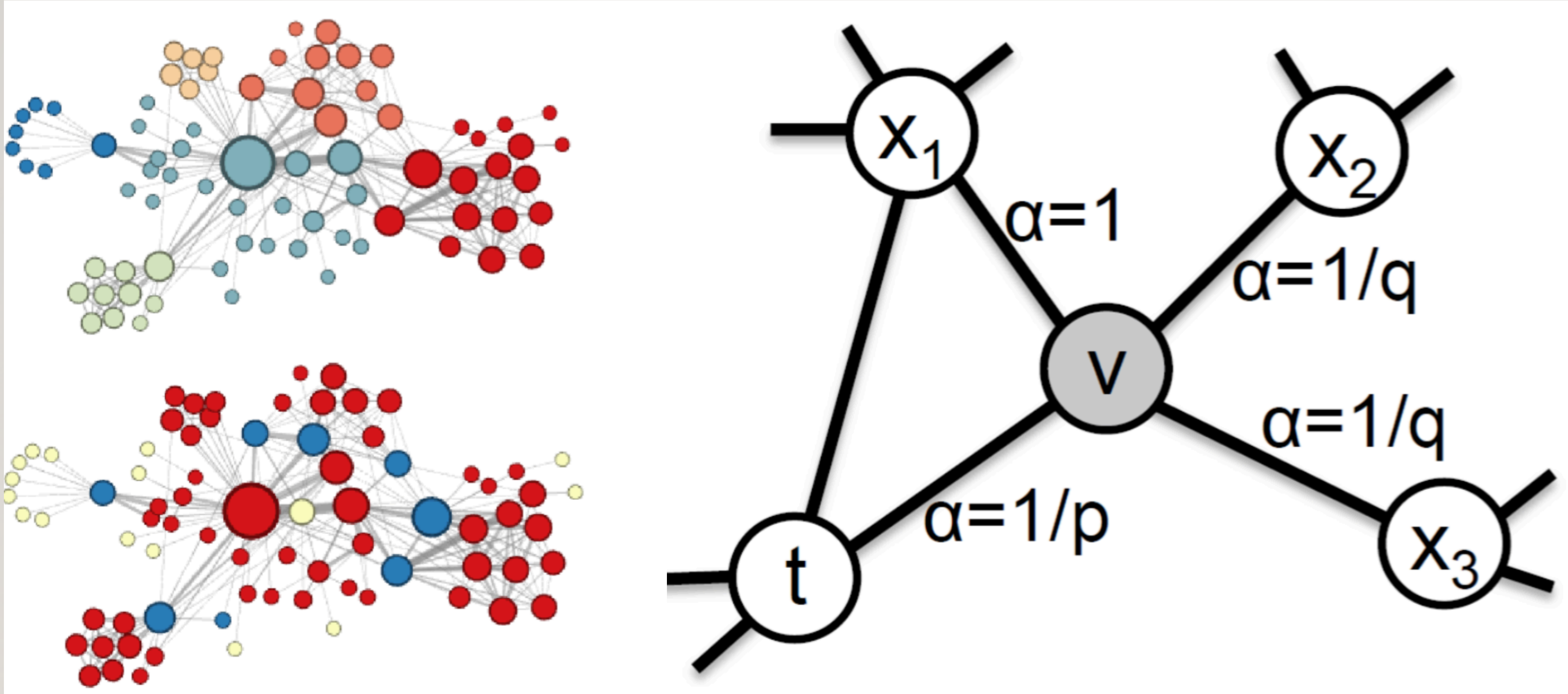
- 二阶马尔科夫过程



- Biased Random Walk: Parameters p, q controls interpolation between DFS and BFS

$$\alpha_{pq}(t, x) = \begin{cases} \frac{1}{p} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases}$$

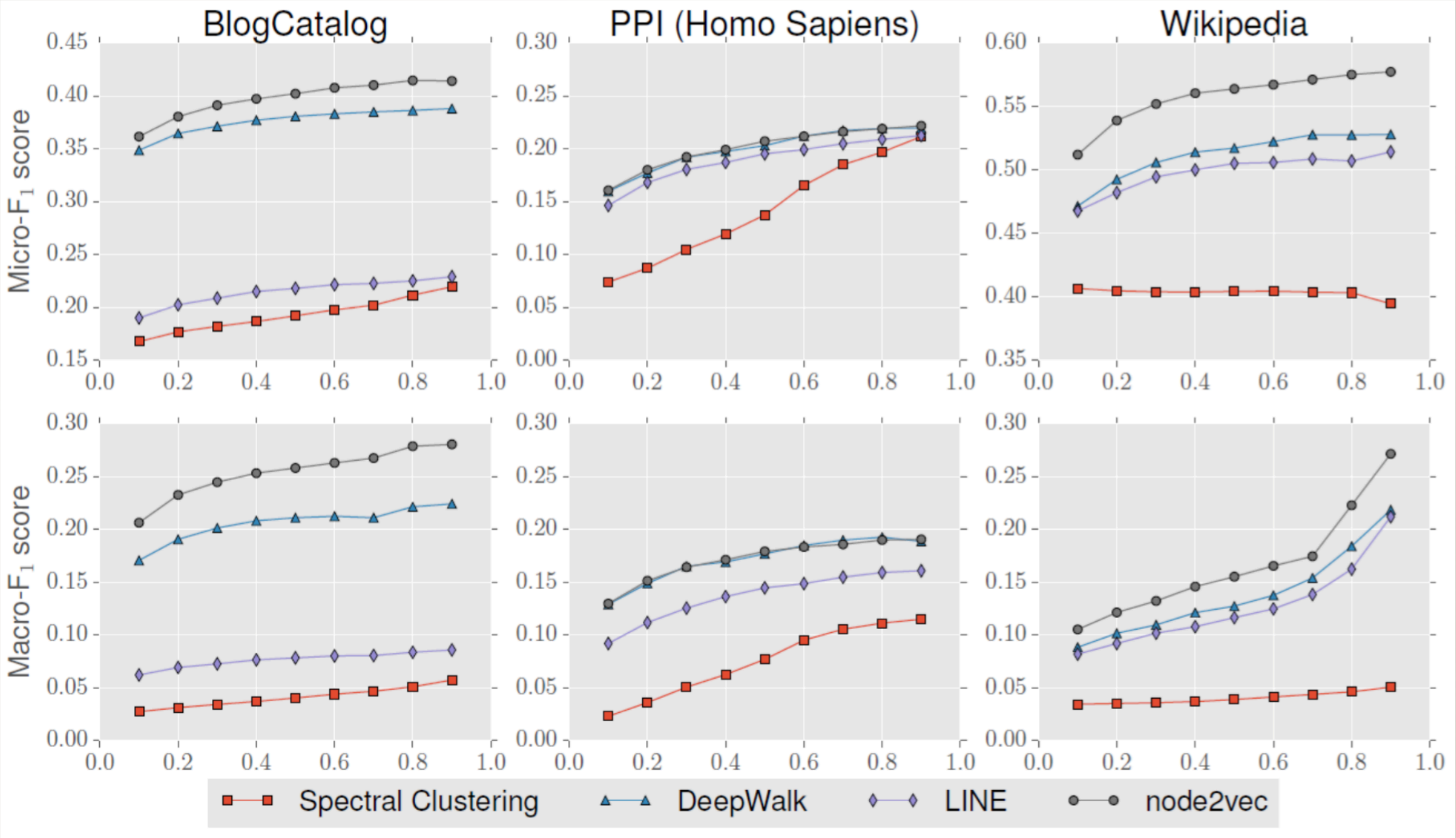
- Social Relations



Homophily (community): $p = 1, q = 2$

Structural equivalence: $p = 1, q = 0.5$

• Experiments - Multi-label classification



- Experiments – Link Prediction

Operator	Symbol	Definition
Average	$\boxplus$	$[f(u) \boxplus f(v)]_i = \frac{f_i(u)+f_i(v)}{2}$
Hadamard	$\boxdot$	$[f(u) \boxdot f(v)]_i = f_i(u) * f_i(v)$
Weighted-L1	$\\|\cdot\\|_{\bar{1}}$	$\\|f(u) \cdot f(v)\\|_{\bar{1}i} = \|f_i(u) - f_i(v)\|$
Weighted-L2	$\\|\cdot\\|_{\bar{2}}$	$\\|f(u) \cdot f(v)\\|_{\bar{2}i} = \|f_i(u) - f_i(v)\|^2$

Table 1: Choice of binary operators $\circ$ for learning edge features. The definitions correspond to the ith component of $g(u, v)$.

- Experiments – Link Prediction

Op	Algorithm	Dataset		
		Facebook	PPI	arXiv
	Common Neighbors	0.8100	0.7142	0.8153
	Jaccard's Coefficient	0.8880	0.7018	0.8067
	Adamic-Adar	0.8289	0.7126	0.8315
	Pref. Attachment	0.7137	0.6670	0.6996
(a)	Spectral Clustering	0.5960	0.6588	0.5812
	DeepWalk	0.7238	0.6923	0.7066
	LINE	0.7029	0.6330	0.6516
	node2vec	0.7266	0.7543	0.7221
(b)	Spectral Clustering	0.6192	0.4920	0.5740
	DeepWalk	**0.9680**	0.7441	0.9340
	LINE	0.9490	0.7249	0.8902
	node2vec	**0.9680**	**0.7719**	**0.9366**
(c)	Spectral Clustering	0.7200	0.6356	0.7099
	DeepWalk	0.9574	0.6026	0.8282
	LINE	0.9483	0.7024	0.8809
	node2vec	0.9602	0.6292	0.8468
(d)	Spectral Clustering	0.7107	0.6026	0.6765
	DeepWalk	0.9584	0.6118	0.8305
	LINE	0.9460	0.7106	0.8862
	node2vec	0.9606	0.6236	0.8477

Table 4: Area Under Curve (AUC) scores for link prediction. Comparison with popular baselines and embedding based methods bootstapped using binary operators: (a) Average, (b) Hadamard, (c) Weighted-L1, and (d) Weighted-L2 (See Table 1 for definitions).

Thanks