

# NODE2VEC

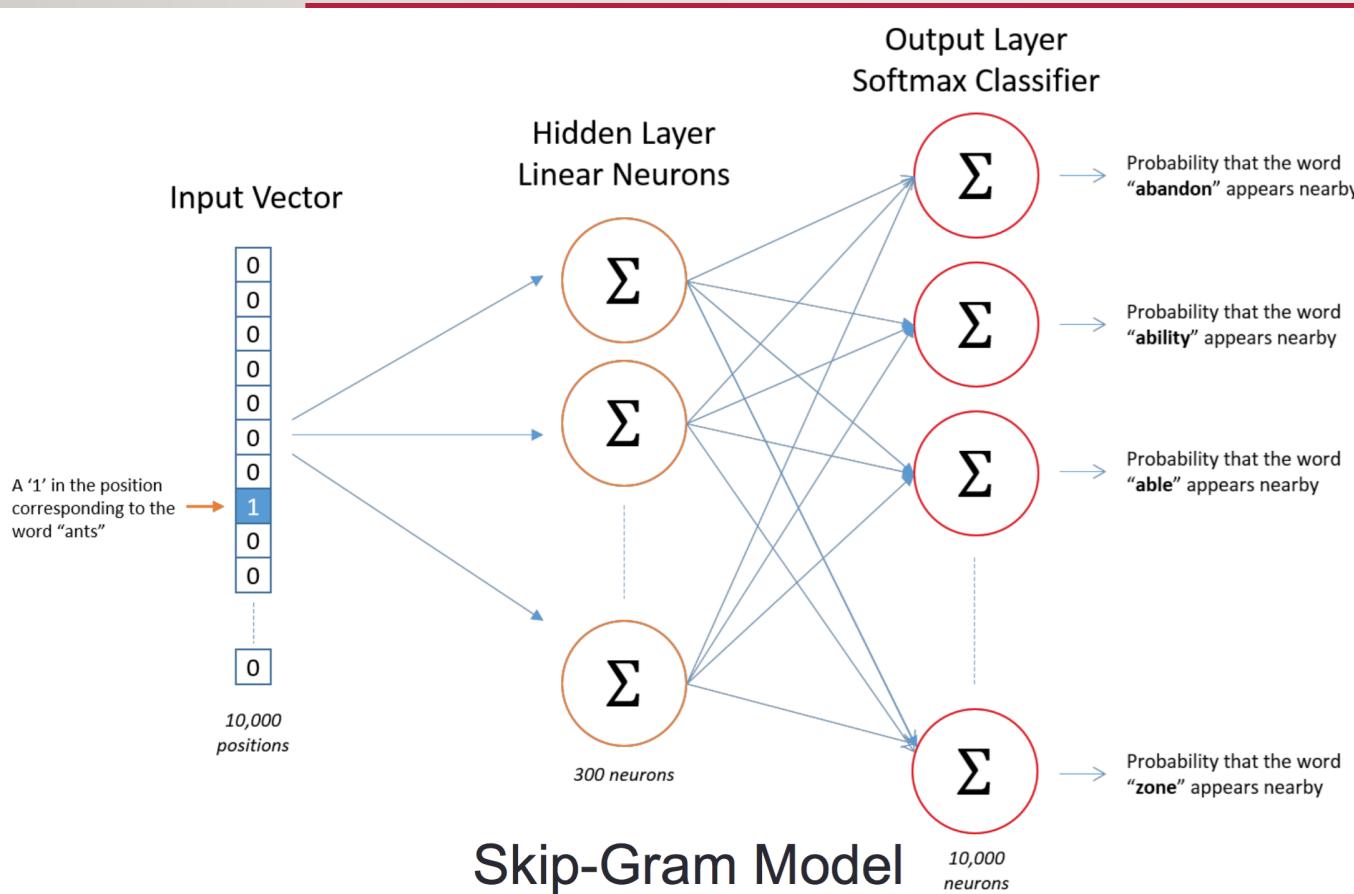
---

KDD 2016

宋军帅

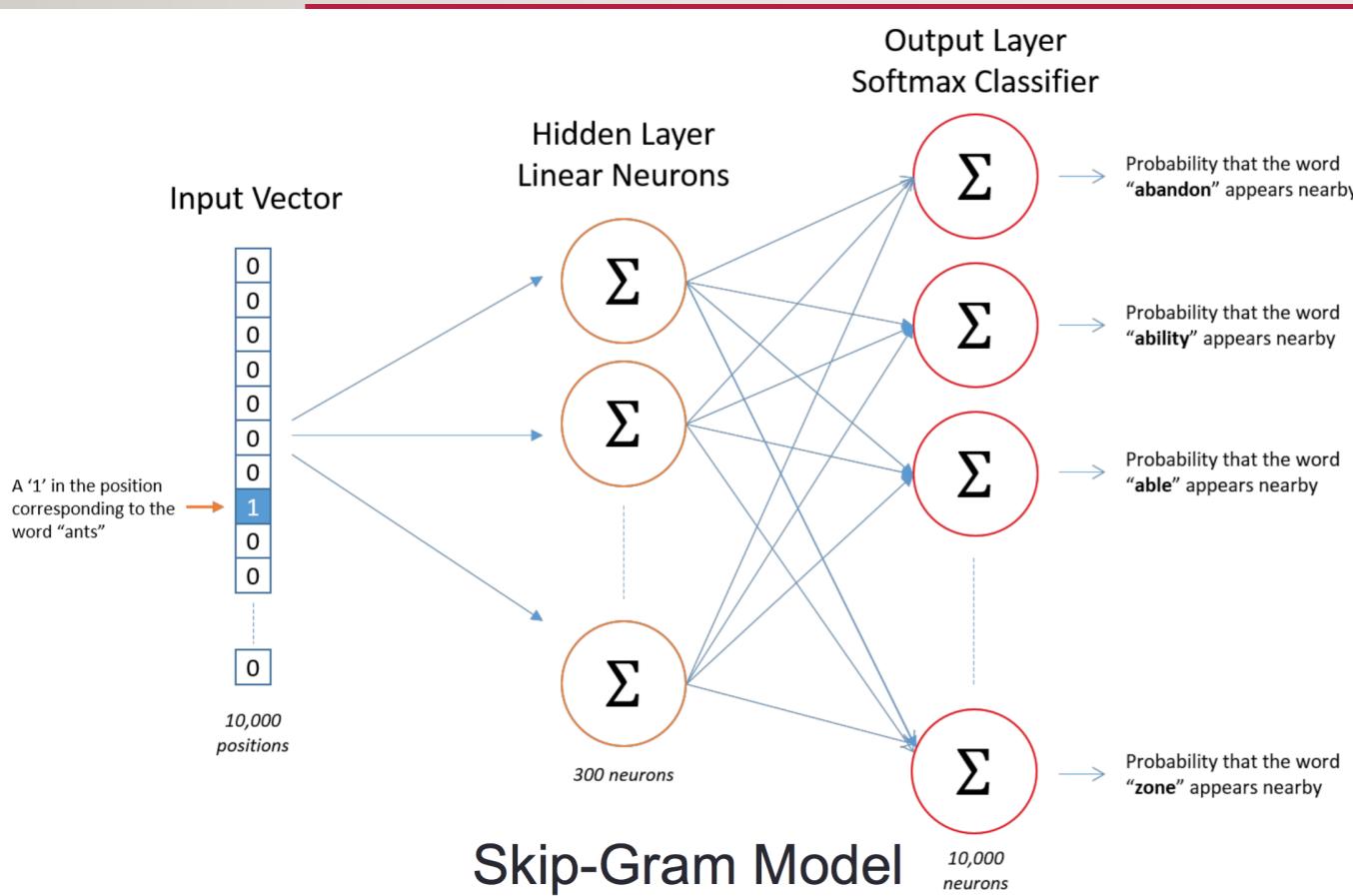
2017/04/11

# 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 $y$
- 真实值为 $Y$
- 我们可以得到单个样本交叉熵损失：

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

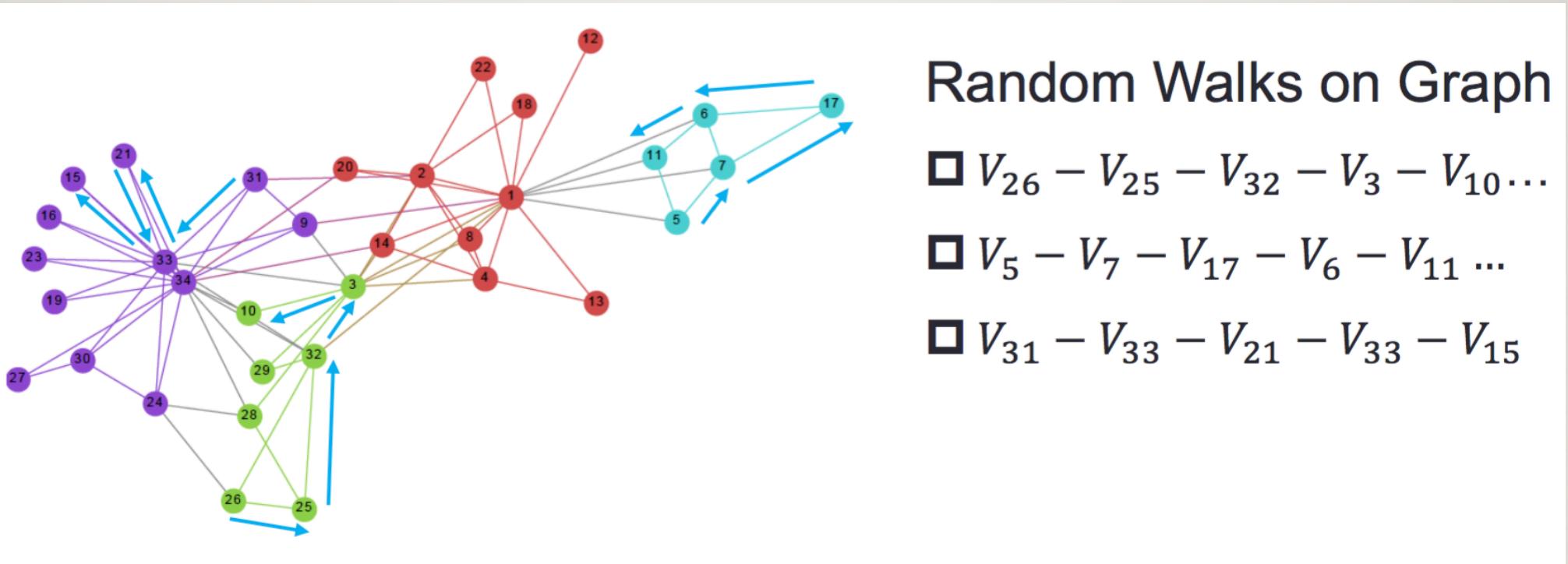
## PRELIMINARY – DEEPWALK

---

- 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

---



## 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  $N_S(u)$  for a node  $u$  conditioned on its feature representation, given by  $f$ :

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

- Conditional independence

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

- Symmetry in feature space

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

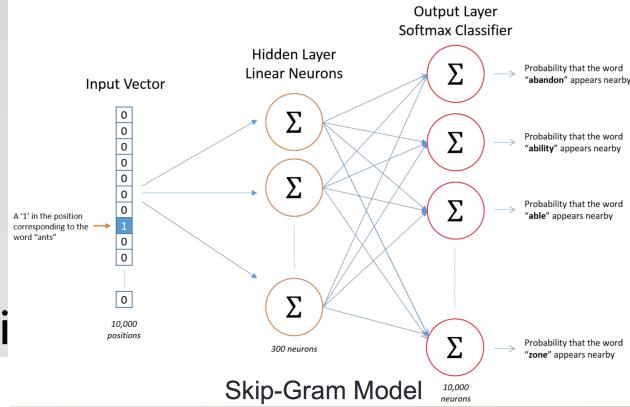
- Objective :

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

- Social Relations

- We seek to optimize the following objective function, which maximizes the probability of observing a network neighborhood  $N_S(u)$  for a node  $u$  conditioned on its 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

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

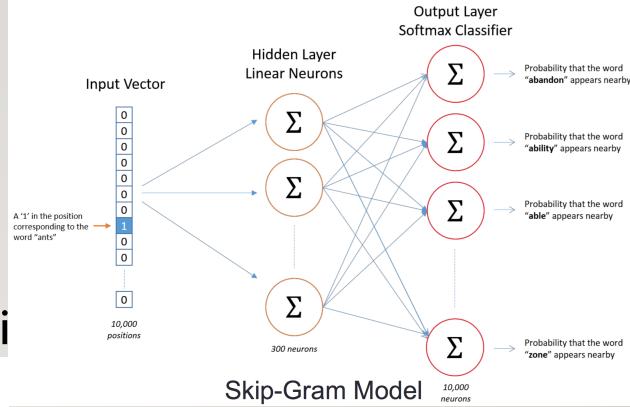
- Objective :

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

- Social Relations

- We seek to optimize the following objective function, which maximizes the probability of observing a network neighborhood  $N_S(u)$  for a node  $u$  conditioned on its 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

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

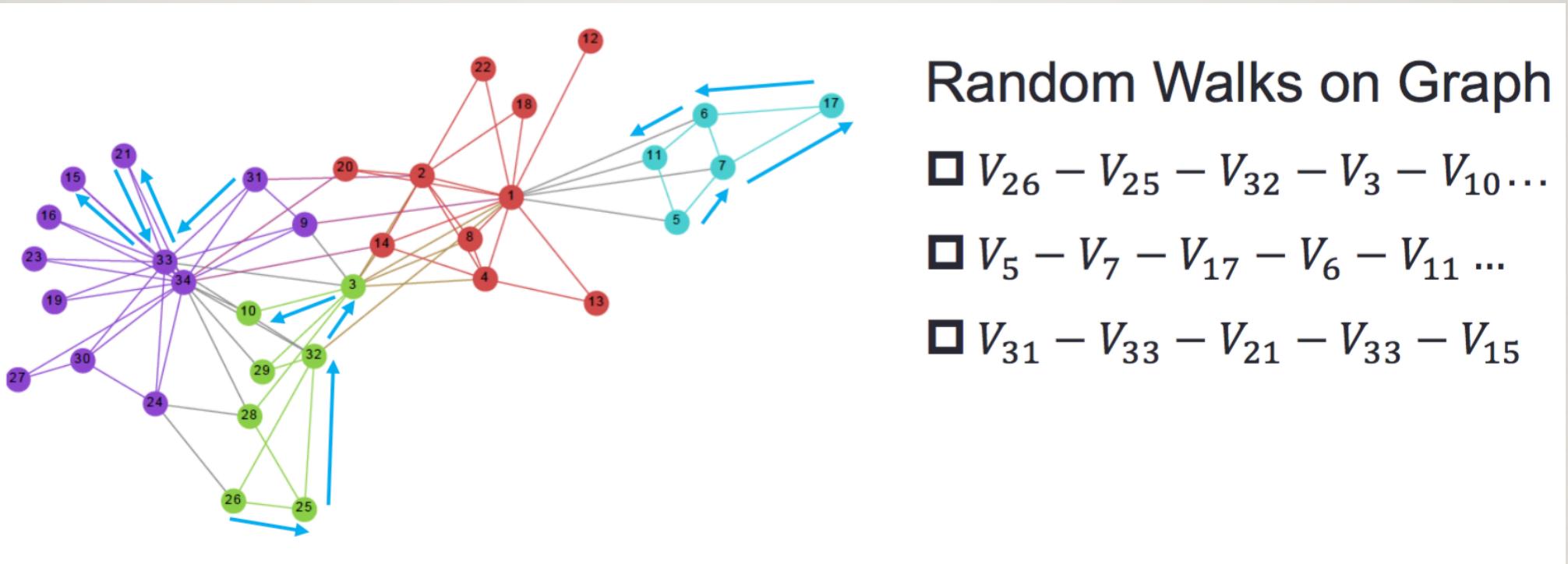
- Objective :

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

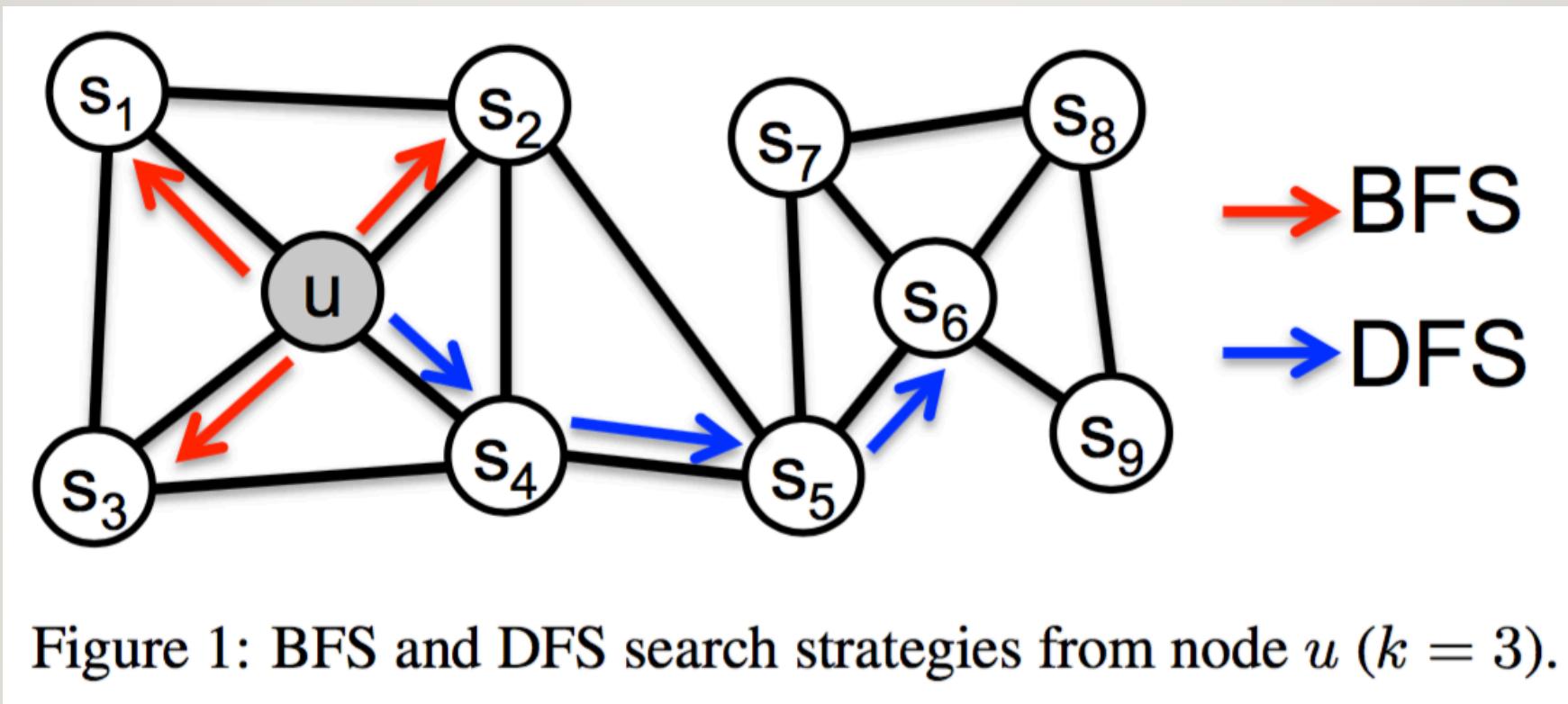
- $N_S(u)$  ?

# PRELIMINARY – DEEPWALK

---



- Node2vec

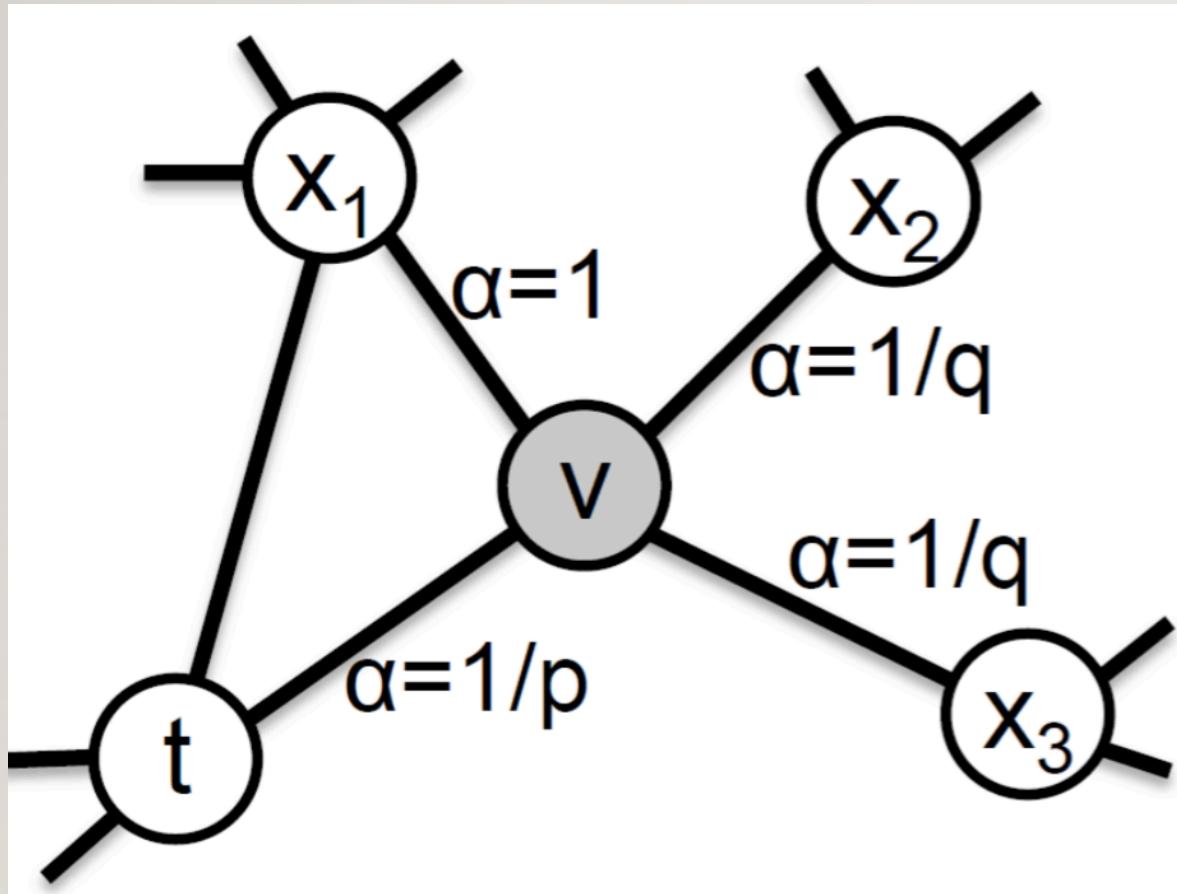


# HOW TO PRESERVE BOTH RELATIONS USING A UNIFIED METHOD?

---

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

- 2<sup>nd</sup> order random walk



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

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

---

**Algorithm 1** The *node2vec* algorithm.

---

**LearnFeatures** (Graph  $G = (V, E, W)$ , Dimensions  $d$ , Walks per node  $r$ , Walk length  $l$ , Context size  $k$ , Return  $p$ , In-out  $q$ )  
 $\pi = \text{PreprocessModifiedWeights}(G, p, q)$   
 $G' = (V, E, \pi)$   
Initialize *walks* to Empty  
**for**  $iter = 1$  **to**  $r$  **do**  
    **for all** nodes  $u \in V$  **do**  
         $walk = \text{node2vecWalk}(G', u, l)$   
        Append *walk* to *walks*  
 $f = \text{StochasticGradientDescent}(k, d, walks)$   
**return**  $f$

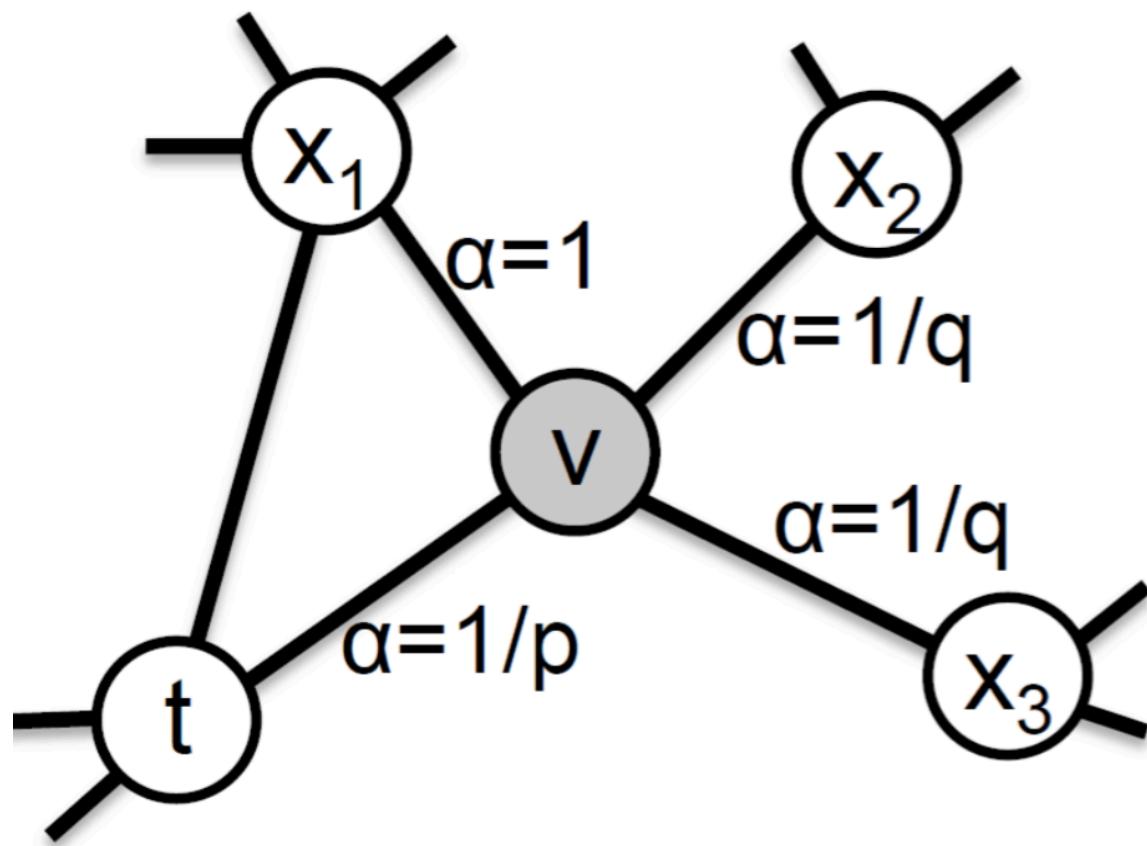
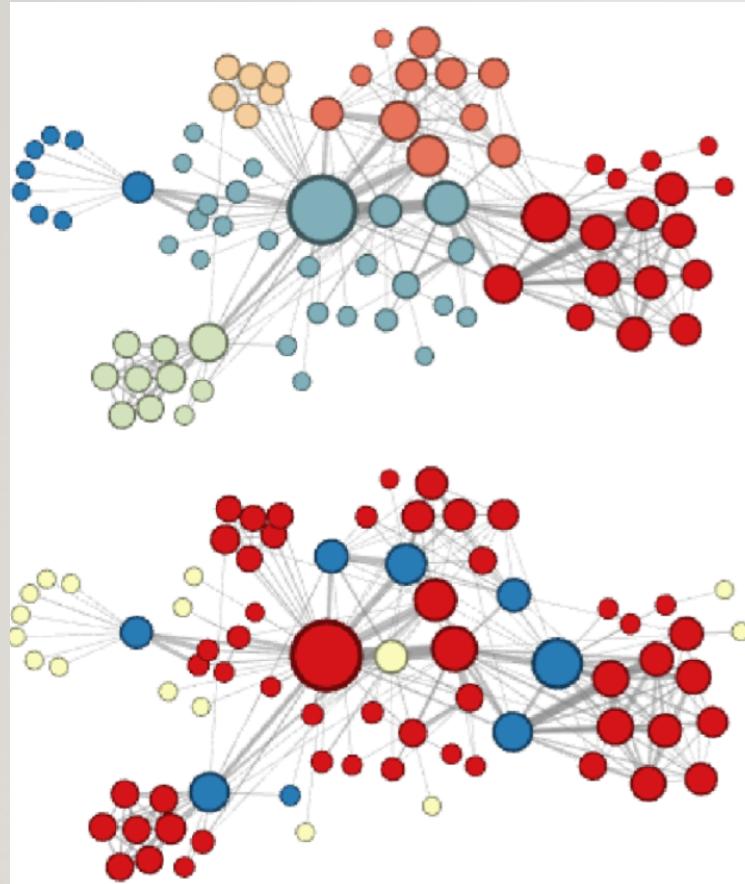
---

**node2vecWalk** (Graph  $G' = (V, E, \pi)$ , Start node  $u$ , Length  $l$ )

Inititalize *walk* to  $[u]$   
**for**  $walk\_iter = 1$  **to**  $l$  **do**  
     $curr = walk[-1]$   
     $V_{curr} = \text{GetNeighbors}(curr, G')$   
     $s = \text{AliasSample}(V_{curr}, \pi)$   
    Append *s* to *walk*  
**return** *walk*

---

- Social Relations



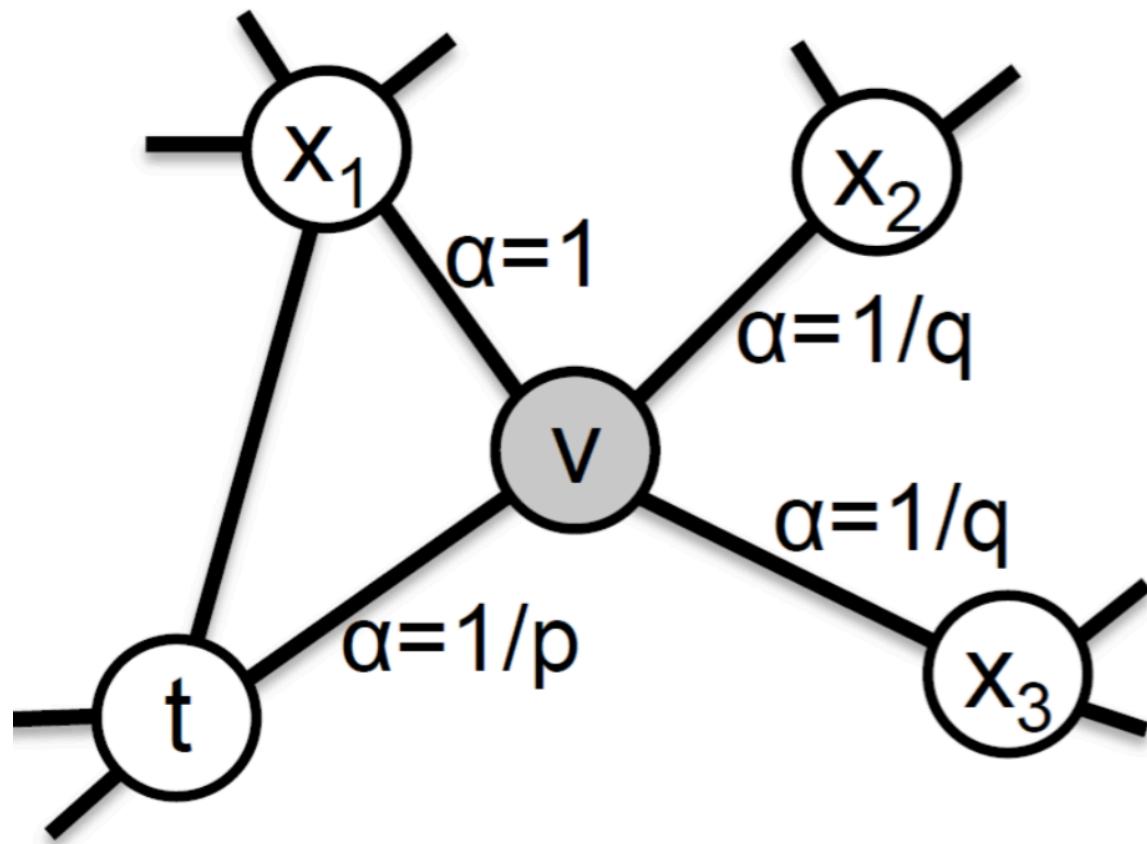
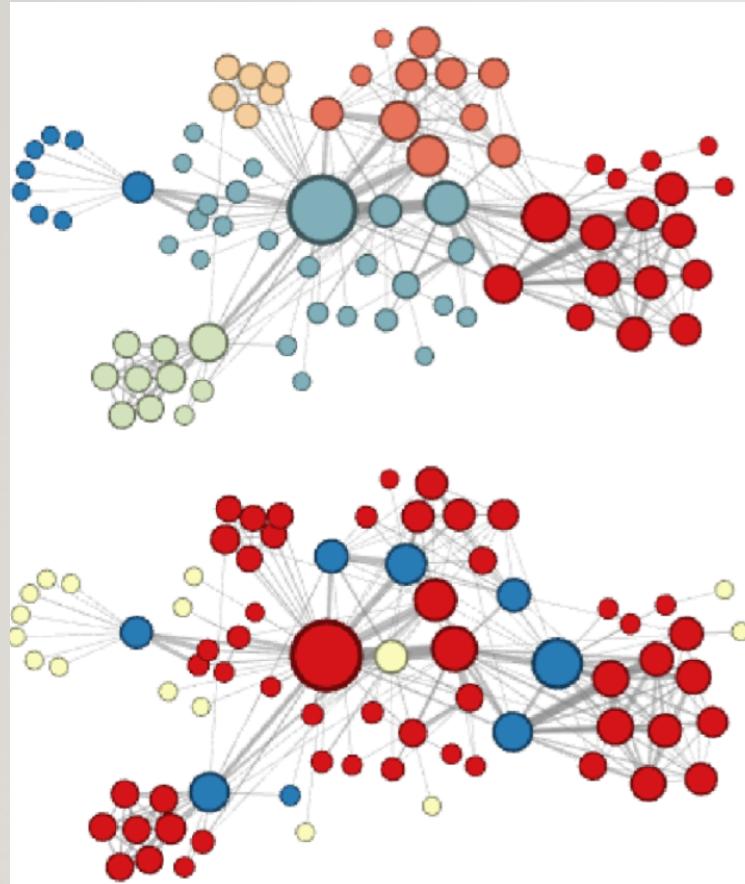
Homophily (community):

$$p = 1, q = 0.5$$

Structural equivalence:

$$p = 1, q = 2$$

- Social Relations



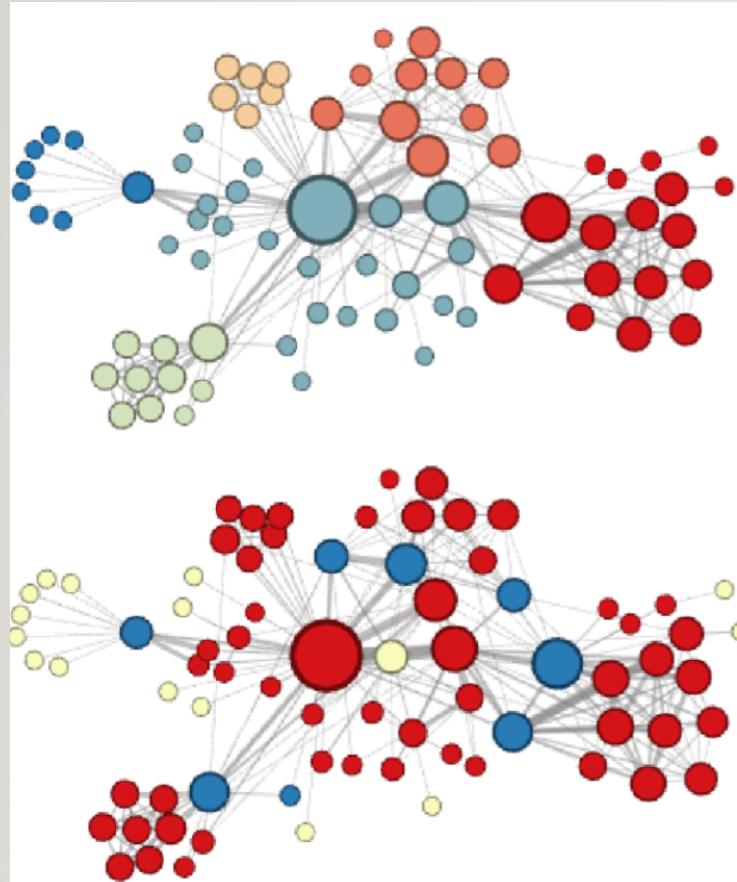
Homophily (community):

$p = 1, q = 0.5$    DFS'

Structural equivalence:

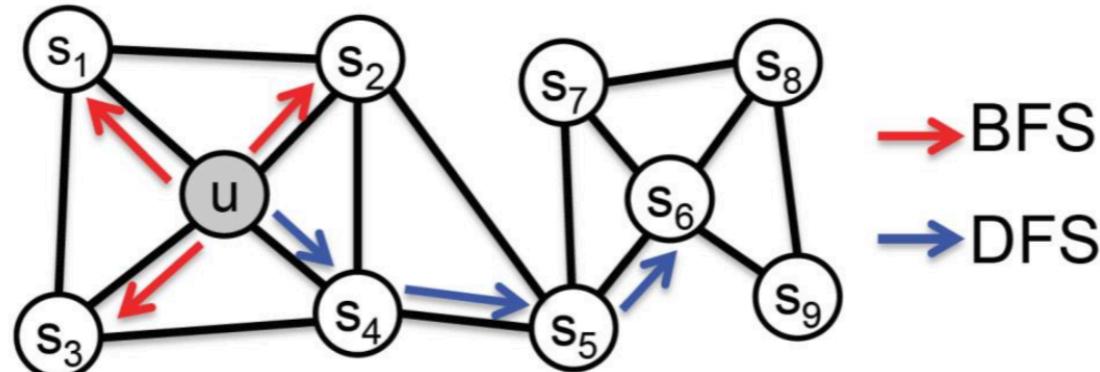
$p = 1, q = 2$    BFS'

- Social Relations



Level 1: BFS + far  
Level 2: BFS + near

Level 3: DFS + near  
Level 4: DFS + far

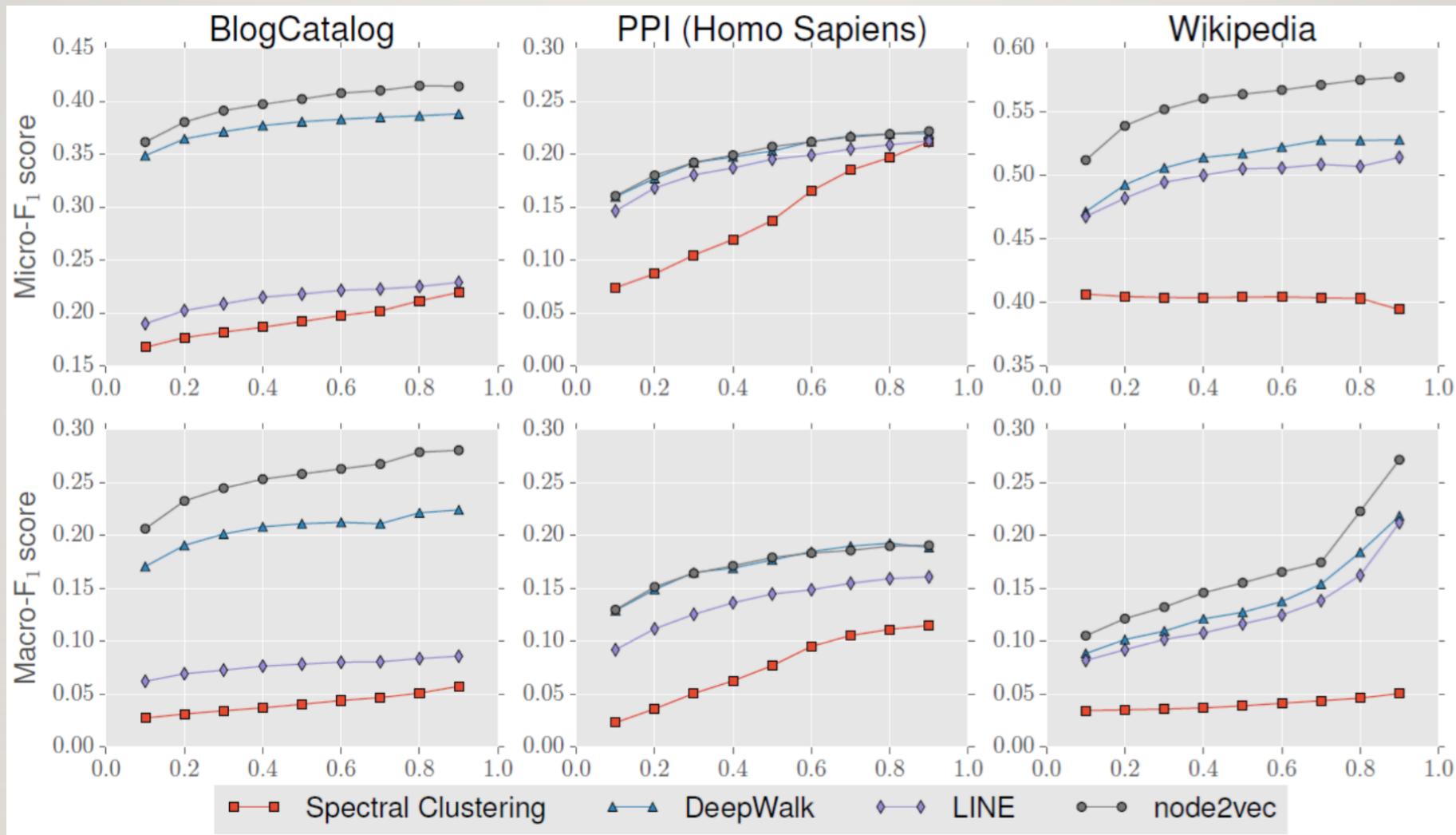


→ BFS  
→ DFS

Homophily (community):  
Structural equivalence:

$p = 1, q = 0.5$     DFS'  
 $p = 1, q = 2$     BFS'

- Experiments - Multi-label classification



# MUTIL-LABEL CLASSIFICATION

---

- one-vs-rest logistic regression classifier.
- the baseline approach, called the *binary relevance* method, amounts to independently training one binary classifier for each label.
- Given an unseen sample, the combined model then predicts all labels for this sample for which the respective classifiers predict a positive result

- Experiments – Link Prediction

<b>Operator</b>	<b>Symbol</b>	<b>Definition</b>
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  $i$ th component of  $g(u, v)$ .

- Experiments – Link Prediction

Op	Algorithm	Dataset		
		Facebook	PPI	arXiv
(a)	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
(b)	Spectral Clustering	0.5960	0.6588	0.5812
	DeepWalk	0.7238	0.6923	0.7066
	LINE	0.7029	0.6330	0.6516
	<i>node2vec</i>	0.7266	0.7543	0.7221
(c)	Spectral Clustering	0.6192	0.4920	0.5740
	DeepWalk	<b>0.9680</b>	0.7441	0.9340
	LINE	0.9490	0.7249	0.8902
	<i>node2vec</i>	<b>0.9680</b>	<b>0.7719</b>	<b>0.9366</b>
(d)	Spectral Clustering	0.7200	0.6356	0.7099
	DeepWalk	0.9574	0.6026	0.8282
	LINE	0.9483	0.7024	0.8809
	<i>node2vec</i>	0.9602	0.6292	0.8468

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

Thanks.