



TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN  
ĐẠI HỌC QUỐC GIA HỒ CHÍ MINH

BÁO CÁO KHÓA LUẬN TỐT NGHIỆP  
**KHUYẾN NGHỊ CỘNG TÁC DỰA TRÊN  
TIẾP CẬN HỌC SÂU**

**GVHD**

: TS.Huỳnh Ngọc Tín

**Sinh viên thực hiện**

: Nguyễn Hữu Đạt

# **NỘI DUNG**

- 1. Đặt vấn đề**
- 2. Phát biểu bài toán**
- 3. Node2Vec**
- 4. Thực nghiệm, đánh giá**
- 5. Kết luận**
- 6. Tài liệu tham khảo chính**

# 1. Đặt vấn đề

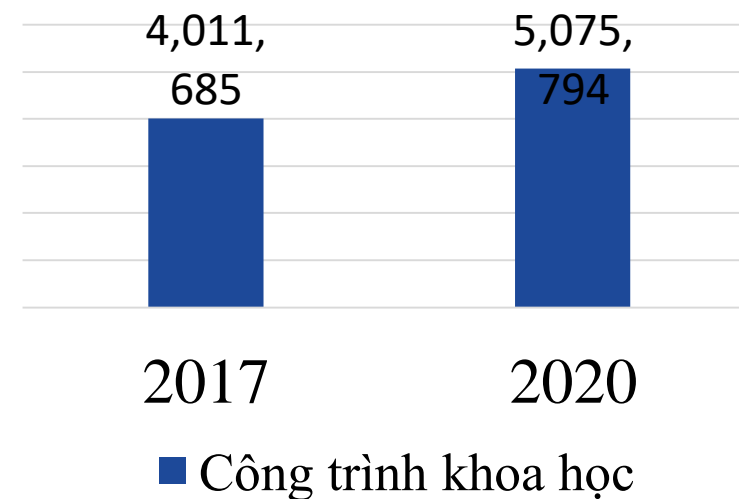
## 1.1. Lợi ích cộng tác nghiên cứu khoa học



**Lợi ích cộng tác trong nghiên cứu khoa học:**  
Thời gian, vốn đầu tư và nhiều chuyên gia  
(Minh họa)

## 1.2. Khó khăn

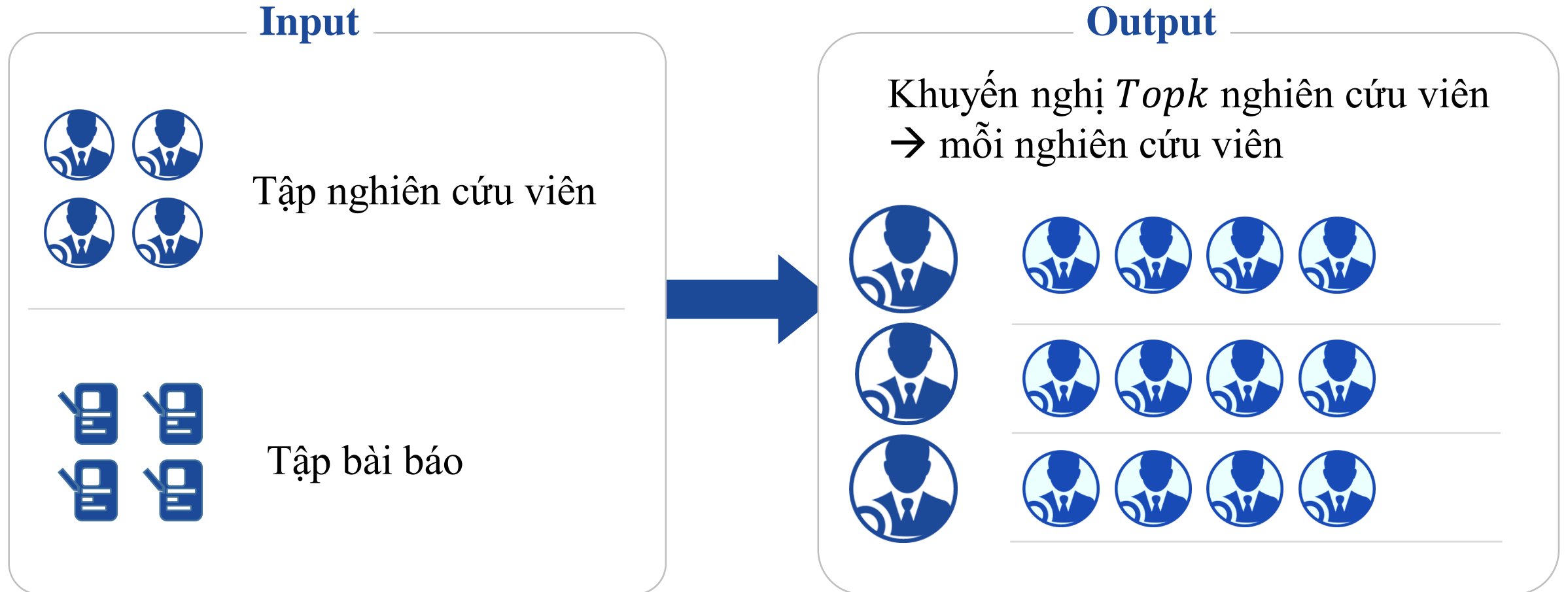
- Số lượng bài báo tăng đột biến
- Số lượng bài báo không lồ



**Sự gia tăng của kho dữ liệu khoa học**  
**dựa trên Cơ sở dữ liệu khoa học DBLP**  
(Nguồn: [DBLP](#), truy cập lần cuối 13/05/2020)

## 2. Phát biểu bài toán

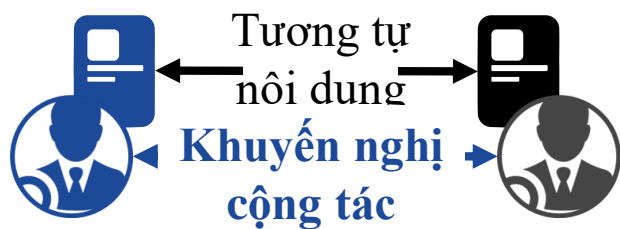
### 2.1. Khuyến nghị cộng tác



## 2. Phát biểu bài toán

### 2.2. Khó khăn, thách thức các tiếp cận phổ biến

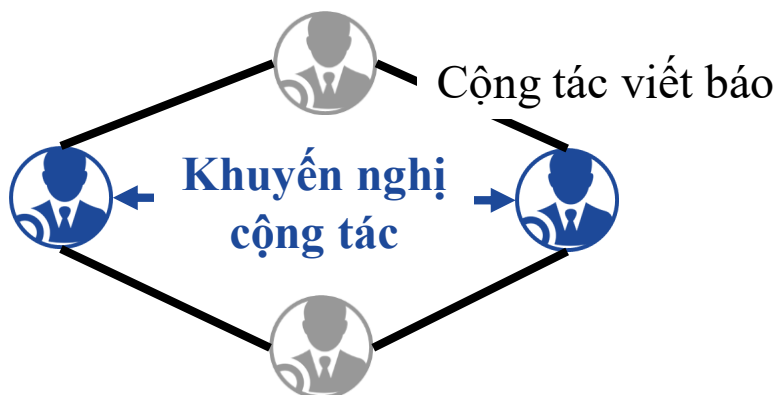
#### 2.2.1 Tiếp cận lọc dựa trên nội dung



- ✗ Tăng số lượng dữ liệu cần xử lý
- ✗ Chỉ có thể xử lý với đối tượng dạng văn bản

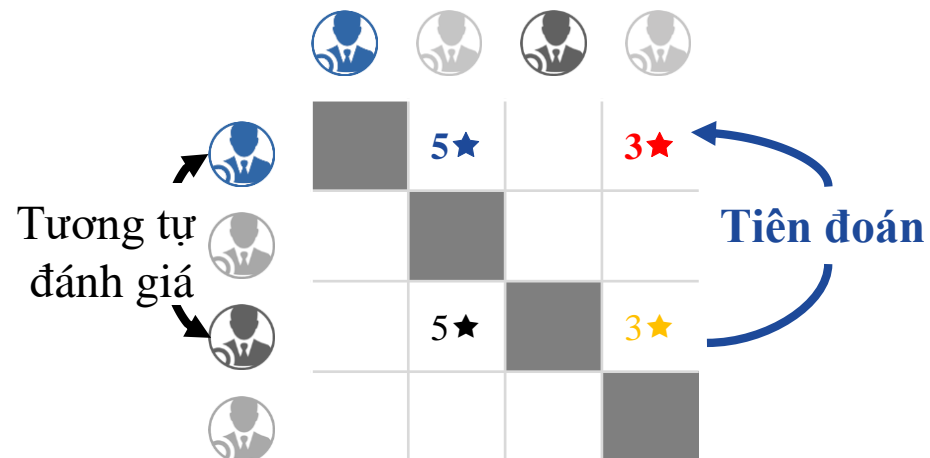
#### 2.2.3 Tương tự đỉnh

Mạng đồng tác giả



- ✗ Không uyển chuyển với mạng đồng tác giả
- ✗ Chỉ dựa trên nguyên tắc liên kết: homophily

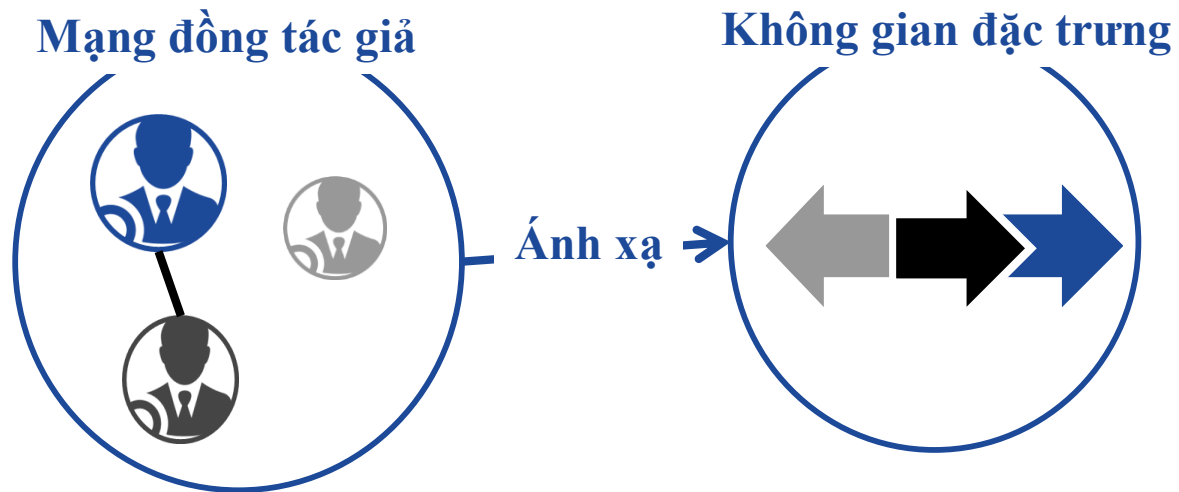
#### 2.2.2 Tiếp cận lọc cộng tác



- ✗ Ma trận đánh giá thưa

## 2. Phát biểu bài toán

### 2.3. Sự khác biệt của Node2vec



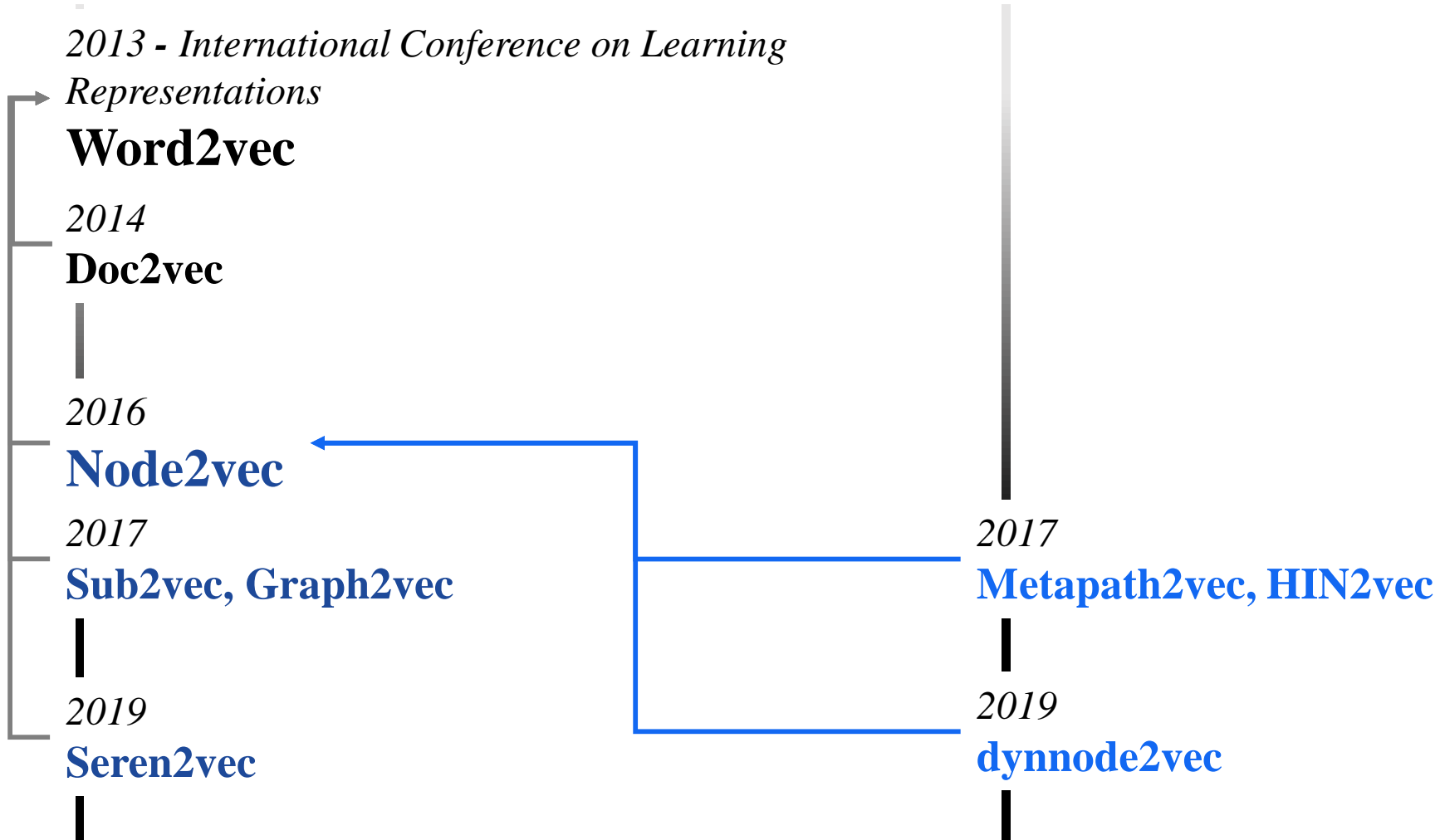
- ✓ Uyển chuyển trong nắm bắt nguyên tắc liên kết
- ✓ Dựa trên 2 nguyên tắc liên kết trong mạng
- ✓ Không cần ~~dữ liệu văn bản~~
- ✓ Không cần ~~ma trận đánh giá~~



## 2. Phát biểu bài toán

### 2.4. Node2vec khởi xướng xu hướng

### 2.5. Node2vec là xu hướng



# 3. Node2Vec

## 3.1. Định nghĩa

“In node2vec, we learn a mapping of nodes to a **low-dimensional space** of **features** that **maximizes the likelihood of preserving network neighborhoods of nodes**.”

– phần Abstract, tài liệu [1]

Mục tiêu: 2 nguyên tắc liên kết

Tài liệu:

[1] [Aditya Grover](#), [Jure Leskovec](#), node2vec: Scalable Feature Learning for Networks [[nguồn](#)][[mã nguồn](#)], Hội nghị [KDD](#), 2016.

[2] [Thomas Mikolov](#) và cộng sự, Efficient Estimation of Word Representations in Vector Space [[nguồn](#)][[mã nguồn](#)], [International Conference on Learning Representations](#), 2013.

## 3.2. Assumption

### Skip-gram model

“..., it tries to maximize classification of a word based on another word in the same sentence” – trang 4, tài liệu [2]



# 3. Node2Vec

## 3.3. Thuật toán

Input:

- **Mạng đồng tác giả**  $G = (U, E)$
- **$d$** : Số chiều của đặc trưng muốn tạo.
- **$l$** : số bước thực hiện [Biased Random-walk](#)
- **$window\_size$** : số nghiên cứu viên lân cận khi thực hiện [Biased Random-walk](#)
- **$p, q$** : Tham số thay đổi probability transition khi thực hiện [Biased Random-walk](#) (bước tiếp, trở về)

Trong đó:

- $U = \{u_1, u_2, \dots, u_N\}$ : Tập nghiên cứu viên.
- $E$  = tập liên kết cộng tác

Output:

- $\forall u \in U$ , khuyến nghị **Topk** nghiên cứu viên cho  $u$

1

**Tìm liên kết của mạng khi bước từ mọi nút**

$\forall u \in U$ ,  $Step_l(u)$  = Tập các nghiên cứu viên bước từ  $u$  trong mạng đồng tác giả = [BiasedRandomWalk](#)( $u, p, q$ )

2

**Ánh xạ từng nút sang không gian đặc trưng**

$\forall u \in U$ ,  $N_s(u)$  = Tập các nghiên cứu viên tương đồng với  $u$ .

Khởi tạo  $f: U \rightarrow \mathbb{R}^d$ .

Tìm  $f$  sao cho  $\max_f \sum_{u \in V} \log(\Pr(N_s(u)|f(u)))$

Trong đó:

- $\Pr(N_s(u)|f(u)) = \prod_{u' \in N_s(u)} \Pr(u'|f(u))$
- $\Pr(u'|f(u)) = \frac{\exp(\text{likelihood}(u', u))}{\sum_{u'' \in V} \exp(\text{likelihood}(u'', u))}$
- $\text{likelihood}(u'', u) = f(u'') \cdot f(u)$

3

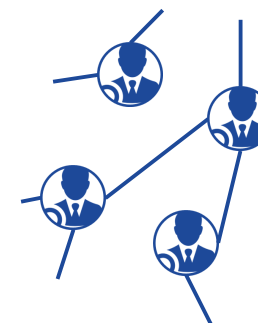
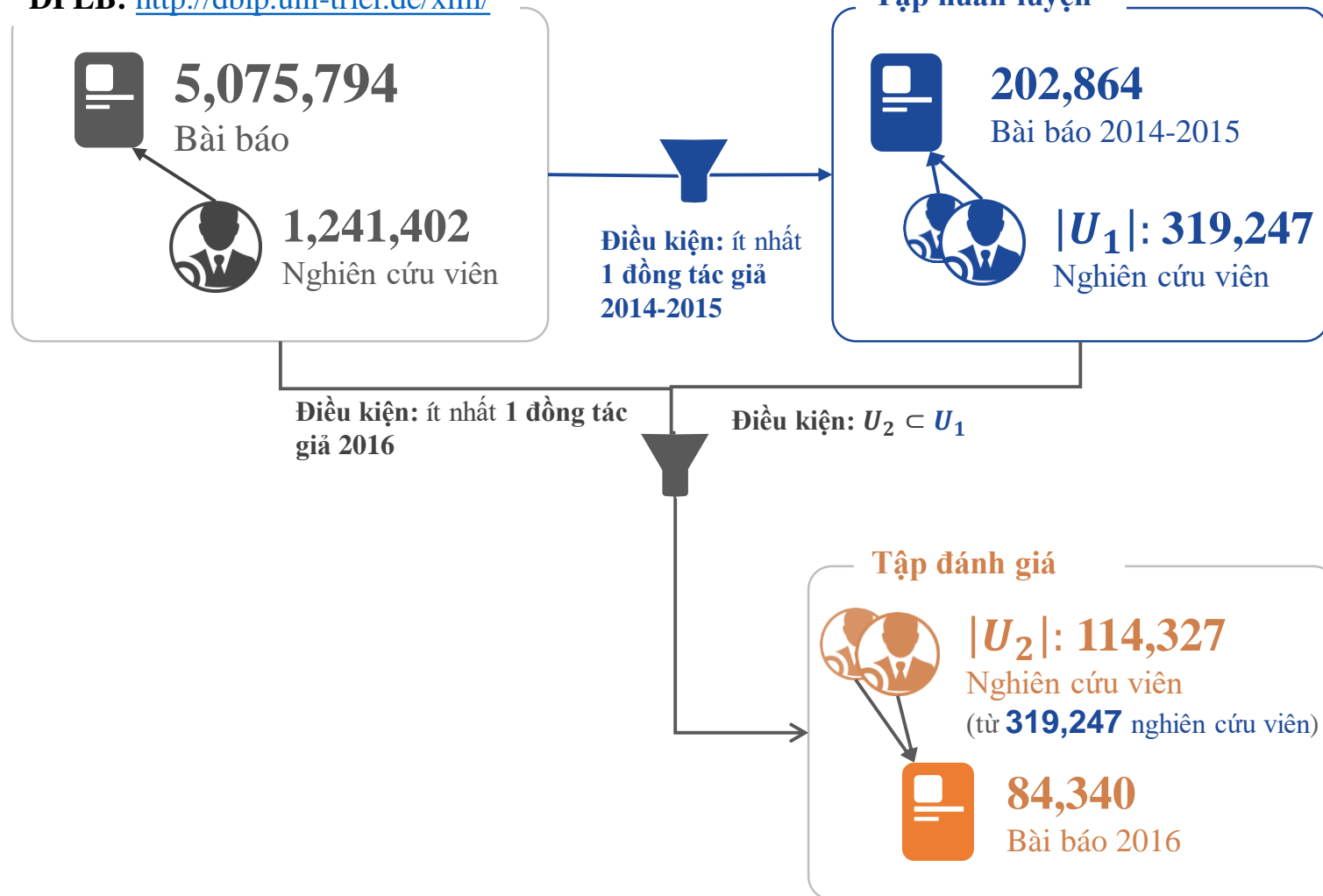
**Khuyến nghị Topk nút có độ tương tự lớn nhất với từng nút trong mạng**

- $\forall u \in U, \forall u' \in U$ , tính độ tương tự theo  $\cosine(f(u), f(u'))$ .
- Thực hiện khuyến nghị TopK

# 4. Thực nghiệm, đánh giá

## 4.1. Dữ liệu thực nghiệm

DPLB: <http://dblp.uni-trier.de/xml/>



**Mạng huấn luyện**

$CoNet_1 = (U_1, Co_1)$

- $|Co_1|$ : 1,042,092 liên kết cộng tác



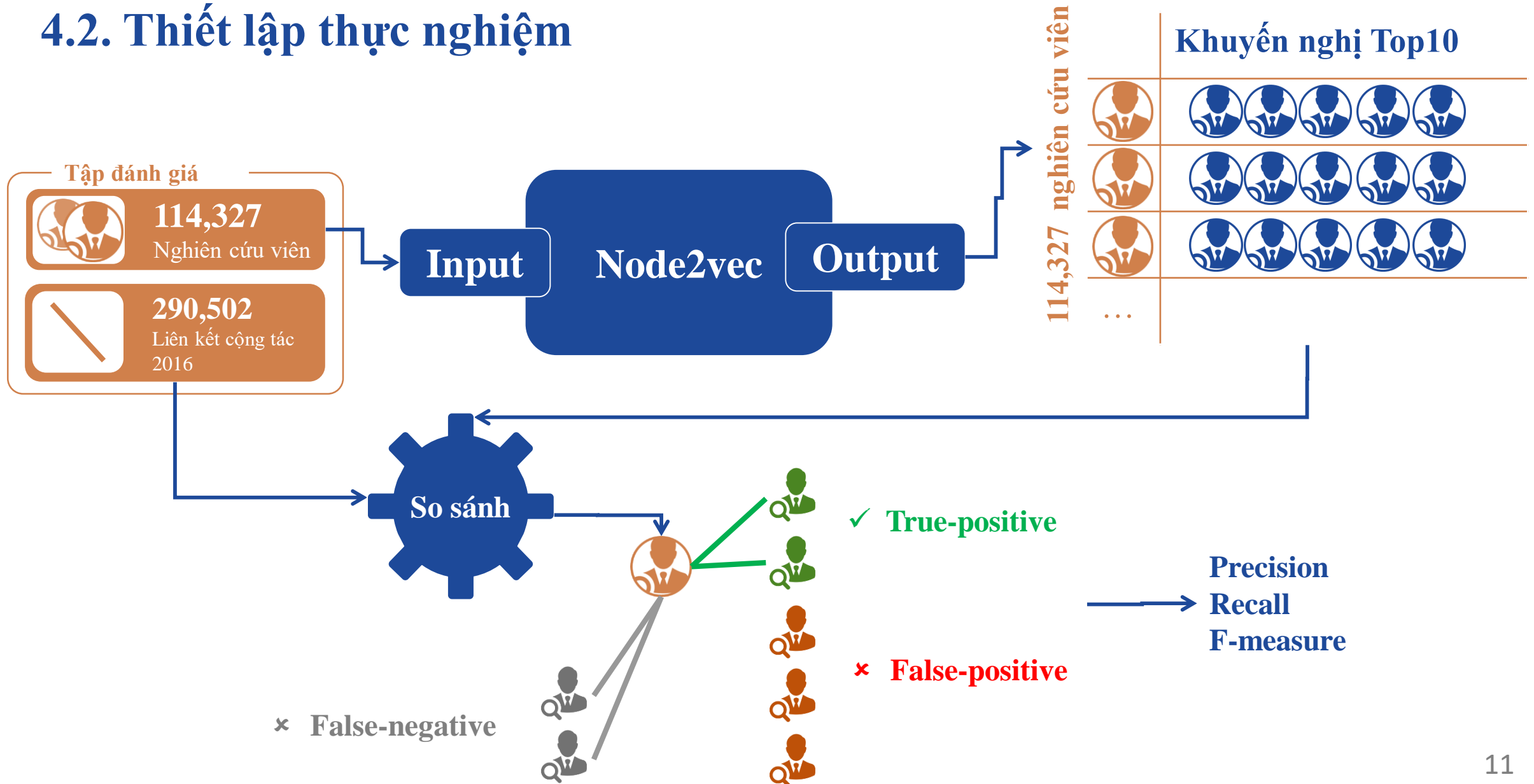
**Mạng đánh giá**

$CoNet_2 = (U_2, Co_2)$

- $|Co_2|$ : 290,502 liên kết cộng tác giữa nghiên cứu viên trong  $U_2$ .

# 4. Thực nghiệm, đánh giá

## 4.2. Thiết lập thực nghiệm



## 4. Thực nghiệm, đánh giá

### 4.3. Phương pháp đánh giá

Độ đo phổ biến trong truy vấn thông tin:

- **Độ chính xác: Precision**  $= P = \frac{tp}{tp+fp}$
- **Độ bao phủ: Recall**  $= R = \frac{tp}{tp+fn}$
- **Độ chính xác trung bình: F – Measure**  $= 2 * \frac{P*R}{P+R}$

Trong đó,

***tp*** (*true positive*): số khuyến nghị đúng.

***fp*** (*true positive*): số khuyến nghị sai.

***fn*** (*false negative*): số chưa được khuyến nghị nhưng đúng

## 4. Thực nghiệm, đánh giá

### 4.4. Kết quả đánh giá

STT	Cách tiếp cận	Precision	Recall	F-Measure
1	<u>Tiếp cận lọc trên nội dung</u> <u>- dựa trên bộ nhớ</u>	<b>0.36</b>	<b>0.35</b>	<b>0.33</b>
2	<u>Common Neighbors</u>	0.19	0.33	0.24
3	<u>Adamic Adar</u>	0.2	0.35	0.25
4	<u>Jaccard Coefficient</u>	0.17	0.29	0.21
5	Node2vec (p=1.5, q=0.5, d=128, num_walks=10, l=80, k=50)	<b>0.17</b>	<b>0.35</b>	<b>0.23</b>

**Mã nguồn:**

**Cấu hình máy:** Máy chủ UIT; RAM: 32G; Cores: 32 - intel(r) xeon(r) cpu e5-2690 v4 @ 2.90ghz  
Bộ nhớ: 300G; Hệ điều hành: Windows Server 2008 R2

# 5. Kết luận



## Vượt trội

- Uyển chuyển và nắm bắt tốt nguyên tắc liên kết trong mạng.
- **Tự động ánh xạ vào không gian vector.**
- **Không cần nội dung bài báo** như tiếp cận lọc nội dung.
- **Không cần ma trận đánh giá** như tiếp cận lọc cộng tác – tận dụng hướng mạng xã hội.

## Thách thức

- Vẫn dựa trên **toàn bộ bộ nhớ** - không thể tiên đoán nghiên cứu viên chưa được học trước.
- Nắm bắt cấu trúc của **mạng hỗn hợp**.
- **Kết quả** tiên đoán chưa đạt được như mong đợi.

## 6. Tài liệu tham khảo chính



- Aditya Grover, Jure Leskovec. node2vec: Scalable Feature Learning for Networks. Stanford university, 2016.
- T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, ‘Distributed Representations of Words and Phrases and their Compositionality’, arXiv:1310.4546 [cs, stat], Oct. 2013, Accessed: May 26, 2020. [Online]. Available: <http://arxiv.org/abs/1310.4546>.
- T. Mikolov, K. Chen, G. Corrado, and J. Dean, ‘Efficient Estimation of Word Representations in Vector Space’, arXiv:1301.3781 [cs], Sep. 2013, Accessed: May 26, 2020. [Online]. Available: <http://arxiv.org/abs/1301.3781>.



**CÁM ƠN THẦY, CÔ HỘI ĐỒNG  
ĐÃ LẮNG NGHE BÀI TRÌNH BÀY CỦA EM**



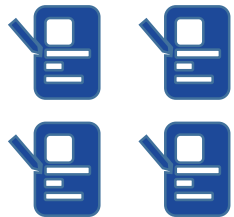
- Đặt vấn đề
  - Tại sao cần khuyến nghị cộng tác ?
- Phát biểu bài toán
  - Khuyến nghị cộng tác là gì ?
  - Khó khăn, thách thức của một số phương pháp phổ biến ?
  - Vì sao dùng Node2vec ?
- Tiếp cận Node2Vec
  - Node2Vec là gì ? Khái niệm liên quan ? Assumption ? (1-2 slides)
  - Chi tiết thuật toán như thế nào ?
- Thực nghiệm, Đánh giá
- Kết luận
- Tài liệu tham khảo chính

# 7. Thuật ngữ liên quan

## 7.1. Mạng đồng tác giả

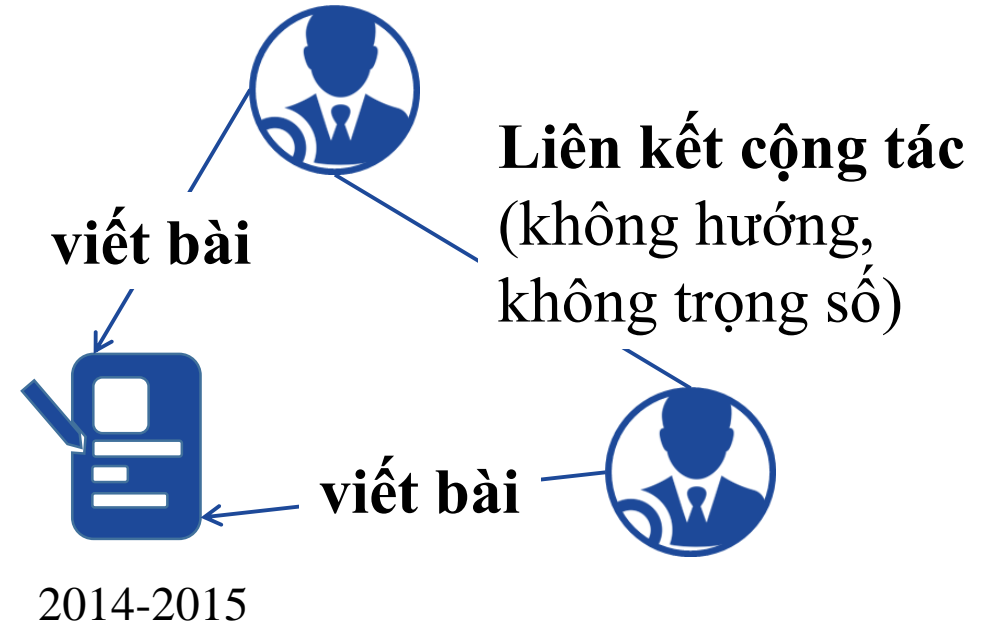


Tập nghiên cứu viên



Tập bài báo

2014-2015





# 7. Thuật ngữ liên quan

## 7.2. Kế thừa Skip-gram, Word2vec

**Output:**

- $\forall u \in U, f: U \rightarrow \mathbb{R}^d$

**Input:**

- $U$  là tập nghiên cứu viên
- $\forall u \in U, \text{Step}_l(u)$  tập bước Biased Random-walk từ nút  $u$  với  $l$  là kích thước tập  $\text{Step}_l(u)$ .
- window\_size.
- $d$ : số chiều của vector đặc trưng.

1

**Khởi tạo**

$\forall u \in U$ , khởi tạo giá trị bất kì cho vector đặc trưng  $f(u)$

2

**Xây dựng Neuron Network**

$u \in U, \forall u' \in \text{Step}_l(u)$ , lấy xây dựng tập nghiên cứu viên lân cận  $N_S(u)$  và xây dựng [Neuron Network](#)

3

**Cập nhật giá trị của từng vector đặc trưng**

Sử dụng [Lan truyền ngược](#) cập nhật vector đặc trưng  $f(u)$

# 7. Thuật ngữ liên quan

C = window\_size × 2

## 7.3. Lan truyền ngược – Neuron Network

Output layer

$\vec{x}$

$x_i$



$N \times 1$  dim

$$W_{delta} = \vec{x}(W_{d \times N} \vec{e})^T$$

$$W_{N \times d} = W_{N \times d} - W_{delta}$$

Hidden layer

$$\vec{h} = W^{N \times d} \vec{x}$$

$h_k$



$d \times 1$  dim

$$W_{d \times N} = \begin{bmatrix} \vec{w}'_1 \\ \dots \\ \vec{w}'_d \end{bmatrix}$$

$$W'_{delta} = \vec{h} \vec{e}^T$$

$$W_{d \times N} = W_{d \times N} - W'_{delta}$$

Output layer

$\vec{y}_1$

$$y_{11} = \text{softmax}(\vec{w}'_1^T \vec{h})$$



$N \times 1$  dim

Round truth



$N \times 1$  dim

Error



$N \times 1$  dim

$\vec{e}$



# 7. Thuật ngữ liên quan

## 7.4. Nguyên tắc liên kết

- **Homophily**

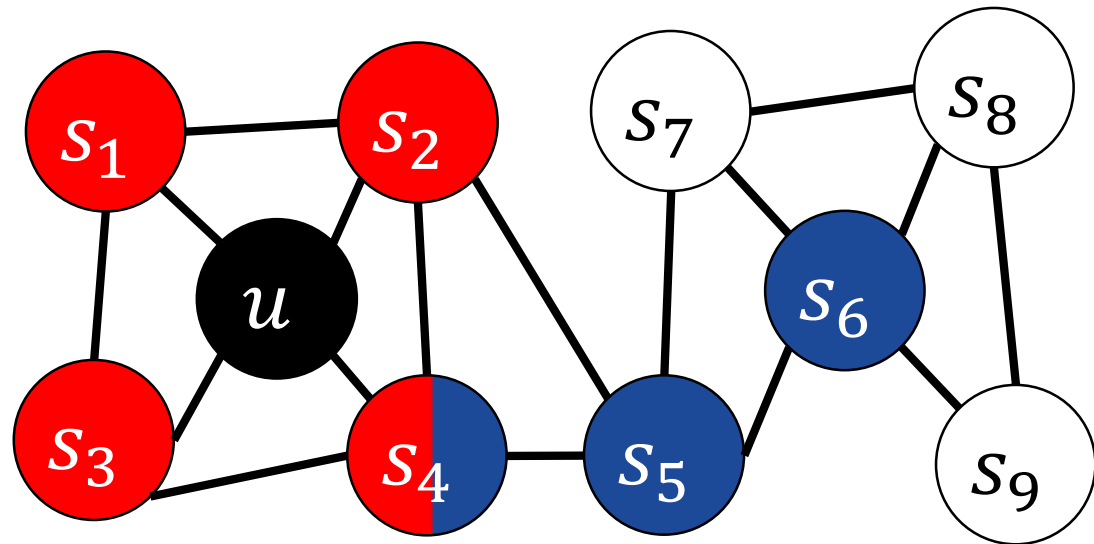
Những nút liên kết trực tiếp với nhau

⇔ gần nhau không gian vector

- **Structural equivalence**

Những nút tương đương cấu trúc

⇔ gần nhau không gian vector



# 7. Thuật ngữ liên quan

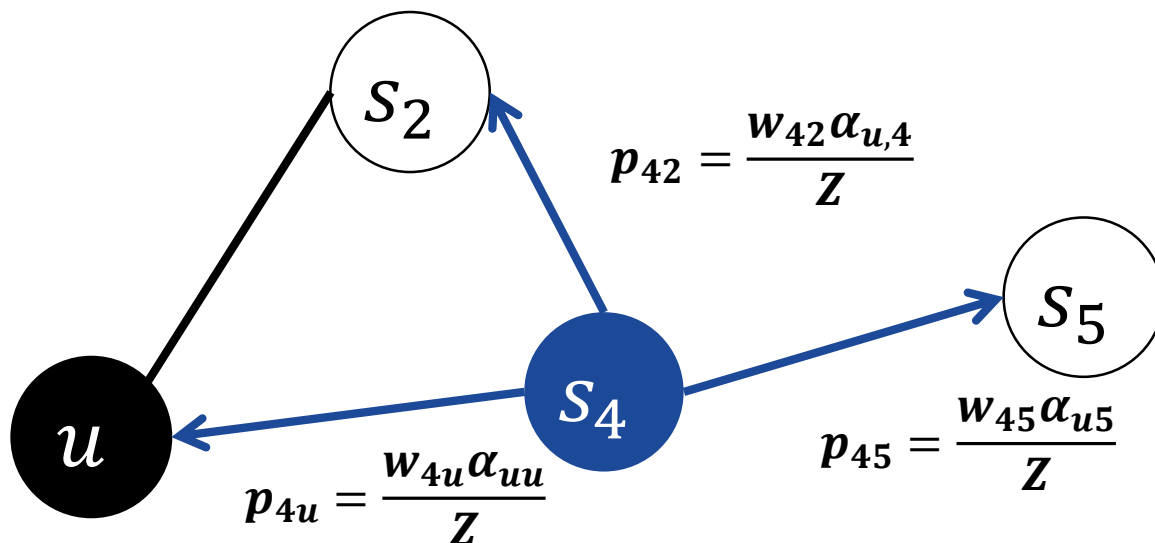
## 7.5. Biased Random-walk

### Lý do

Vì bước đi để tìm lân cận theo 2 định nghĩa: Homophily và Structural equivalence

### Thực hiện

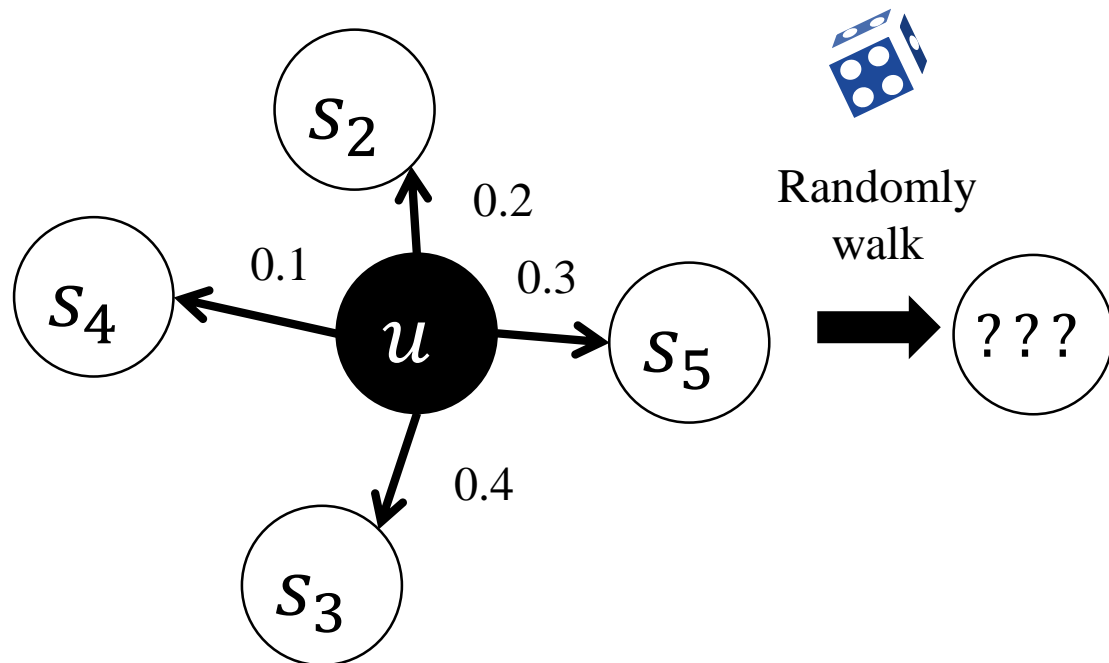
- 1 Tại mỗi nút, lưu trữ bảng random-walk tới những nút liên kề với nút đó.
- 2  $l \in \mathbb{N}$ ,  $\mathbf{Step}_l(u) = \{\mathbf{u}\}$  While  $|\mathbf{Step}_l(u)| < l$ ,  $\mathbf{Step}_l(u) = \text{RandomWalk}(u, \mathbf{Step}_l(u)[-1]) \cap \mathbf{Step}_l(u)$
- 3  $|N_S(u)| = k, k < l$



$$\alpha_{u,x} = \begin{cases} \frac{1}{p} & \text{Nếu } d_{u,x} = 0 \\ \frac{1}{q} & \text{Nếu } d_{u,x} = 1 \\ \frac{1}{q} & \text{Nếu } d_{u,x} = 2 \end{cases}$$

# 7. Thuật ngữ liên quan

## 7.6. Bảng random-walk truyền thống:

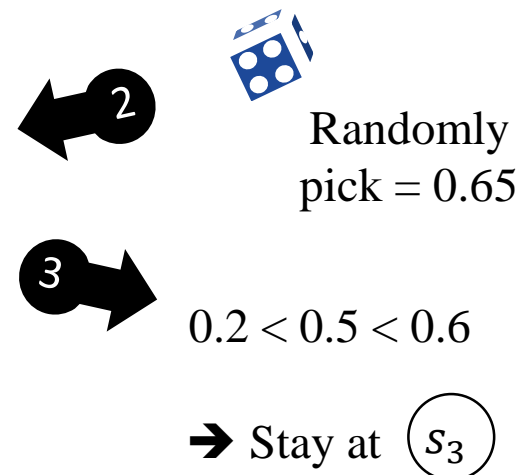


Random  
generate

	$u$
$s_2$	0.2
$s_3$	0.4
$s_4$	0.1
$s_5$	0.3

Giả sử:

	$u$
$s_2$	0.2
$s_3$	0.6
$s_4$	0.7
$s_5$	1



Độ phức tạp mỗi walk =  $O(|U| - 1)$

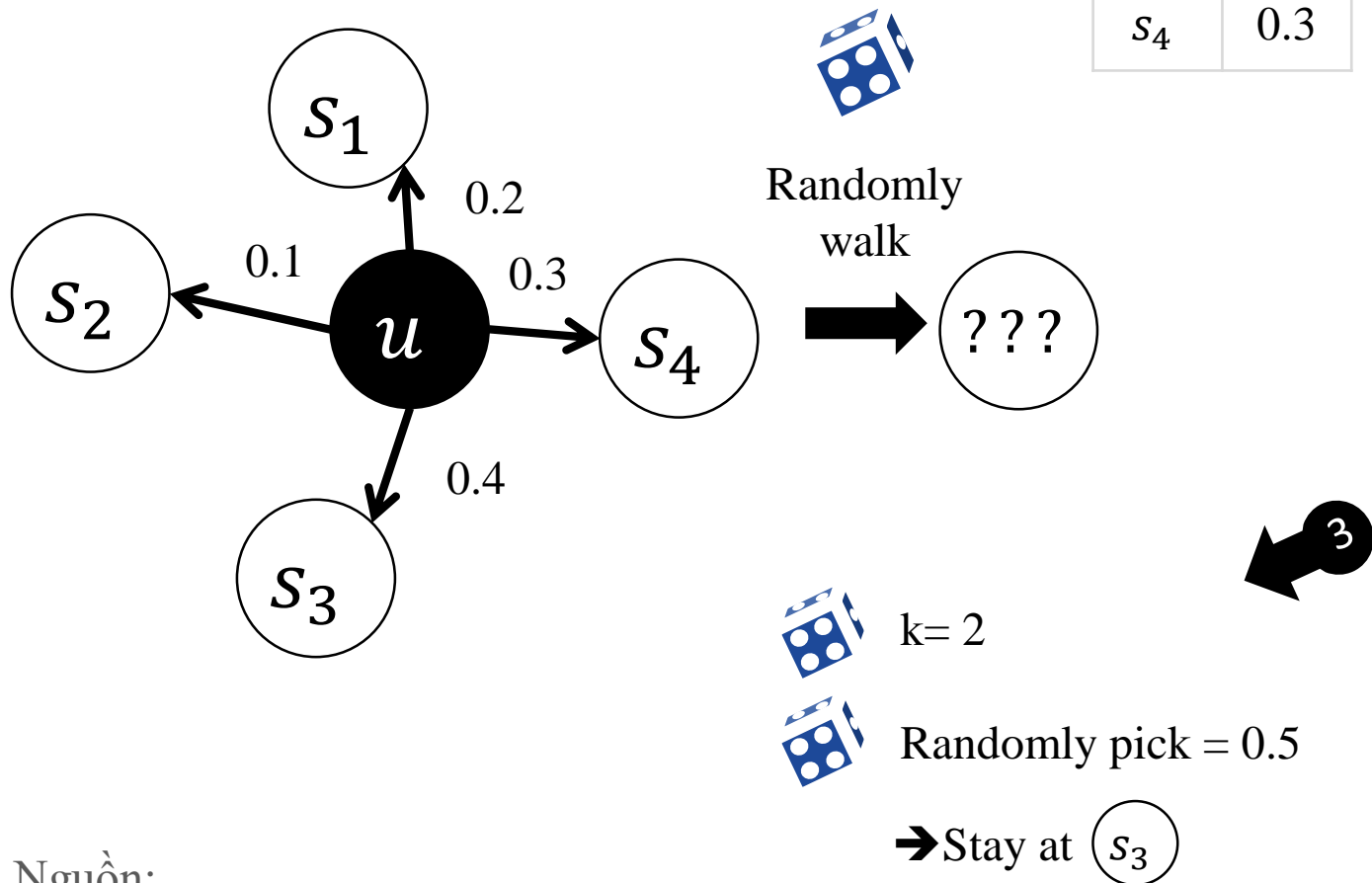
Nguồn:

[1] <https://lips.cs.princeton.edu/the-alias-method-efficient-sampling-with-many-discrete-outcomes/>

[2] <http://cgi.cs.mcgill.ca/~enewel3/posts/alias-method/index.html>

# 7. Thuật ngữ liên quan

## 7.6. Bảng random-walk cải tiến:



	$u$
$s_1$	0.2
$s_2$	0.1
$s_3$	0.4
$s_4$	0.3

1 →

walk			
1	0.25	$s_1$ 0.20	$s_4$ 0.05
2	0.50	$s_2$ 0.10	$s_3$ 0.15
3	0.75	$s_3$ 0.25	$s_1$ 0
4	1	$s_4$ 0.25	$s_1$ 0

2 ↙

walk			
1	0.25	$s_1$ 0.8	$s_4$ 1
2	0.50	$s_2$ 0.4	$s_3$ 1
3	0.75	$s_3$ 1	$s_1$ 0
4	1	$s_4$ 1	$s_1$ 0

3 ↙

Độ phức tạp mỗi walk =  $O(1)$

Nguồn:

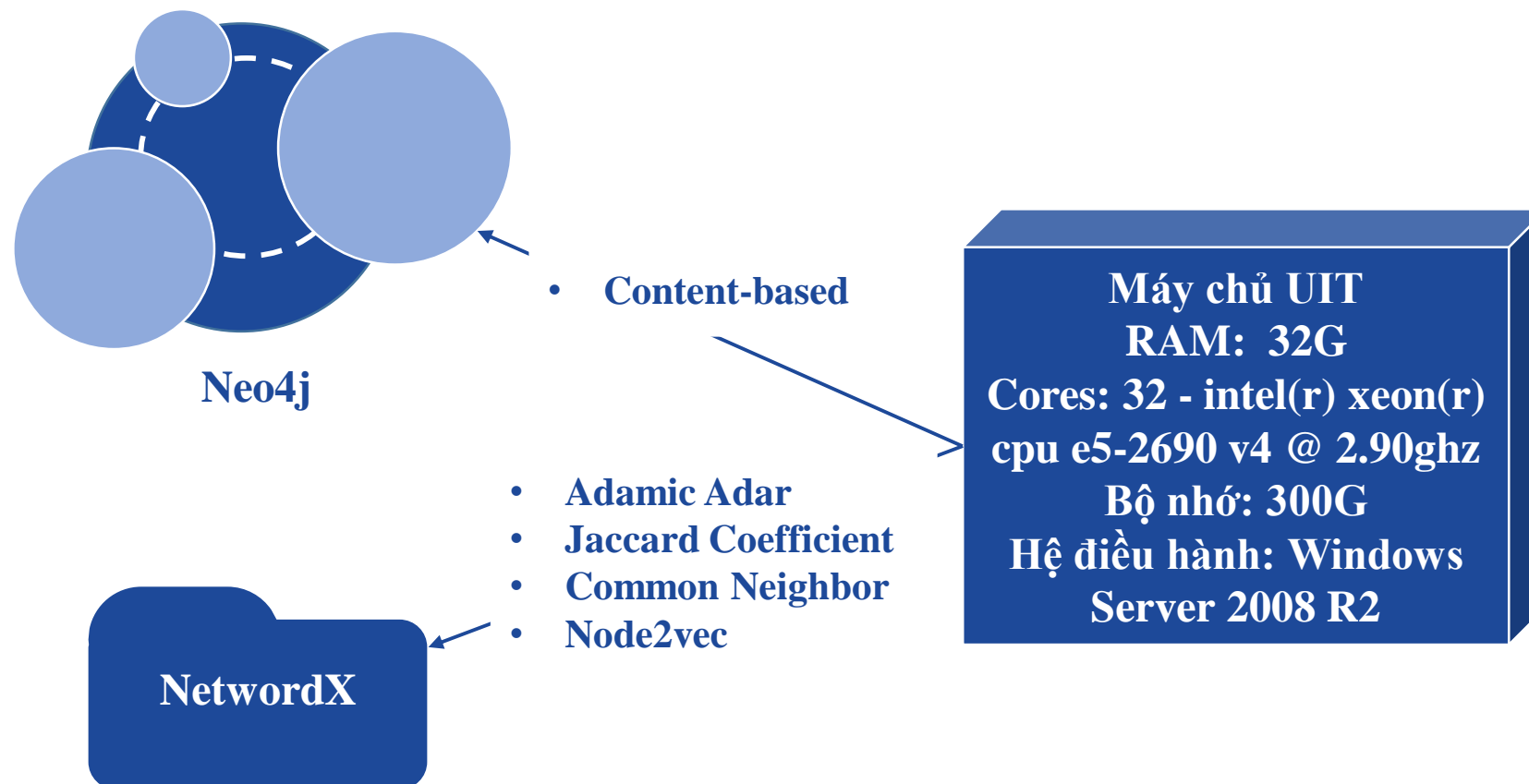
[1] <https://lips.cs.princeton.edu/the-alias-method-efficient-sampling-with-many-discrete-outcomes/>

[2] <http://cgi.cs.mcgill.ca/~enewel3/posts/alias-method/index.html>



# 8. Thiết lập

## 8.1. Sơ đồ triển khai



# 8. Phương pháp phổ biến

## 8.2. Lọc dựa trên nội dung

**Input:**  $U$ : Tập nghiên cứu viên,  $P$ : Tập bài báo

**Output:**

$\forall u \in U$ , danh sách TopK nghiên cứu viên dựa trên giá trị hàm hữu ích  $f$

1  $Content(p) = \vec{w}'_p = (w_1, w_2, \dots, w_V)$



$p_v$



$\vec{w}_p = (w_1, w_2, \dots, w_V)$



2  $UserProfile(u) = \vec{w}'_u = (w'_1, w'_2, \dots, w'_V)$



Nghiên cứu viên

Bài báo công bố



Hồ sơ người dùng



3  $\forall u \in U, \forall u' \in U, f(u', u) = \text{Cosine}(\vec{w}_u, \vec{w}_{u'}) = \frac{\vec{w}_u \vec{w}_{u'}^T}{\|\vec{w}_u\| * \|\vec{w}_{u'}\|}$

4  $\forall u \in U$ , khuyến nghị TopK nghiên cứu viên dựa trên giá trị hàm hữu ích  $f$

# 8. Phương pháp phổ biến

## 8.2. Tương tự đỉnh

**Input:** CoNet: Mạng đồng tác giả

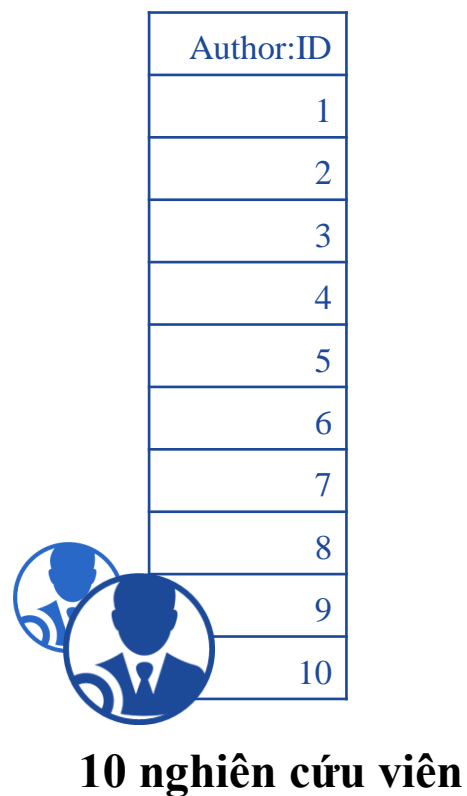
**Output:**  $\forall u \in U$ , danh sách TopK  
nghiên cứu viên dựa trên giá  
trị hàm hữu ích  $f$

- 1  $\forall u \in U, \forall u' \in U, f(u', u) =$ 
  - **Common Neighbors:**  $|U_u \cap U_{u'}|$
  - **Jaccard Coefficient:**  $\frac{|U_u \cap U_{u'}|}{|U_u \cup U_{u'}|}$
  - **Adamic Adar:**  $|U_u \cap U_{u'}|$
- 2  $\forall u \in U$ , khuyến nghị TopK nghiên cứu viên dựa trên giá trị hàm hữu ích  $f$

# 9. Ví dụ Node2vec

## 9.1. Giả thiết:

Liên kết cộng tác viết báo khoa học



start:ID	end:ID
5	1
2	3
10	6
5	2
3	4
2	9
8	7
6	7
7	9


start:ID	end:ID
5	2
5	8
2	7
2	9
3	6
4	10
6	10



# 9. Ví dụ Node2vec


## 9.2. Mạng đồng tác giả:

### Ghi chú

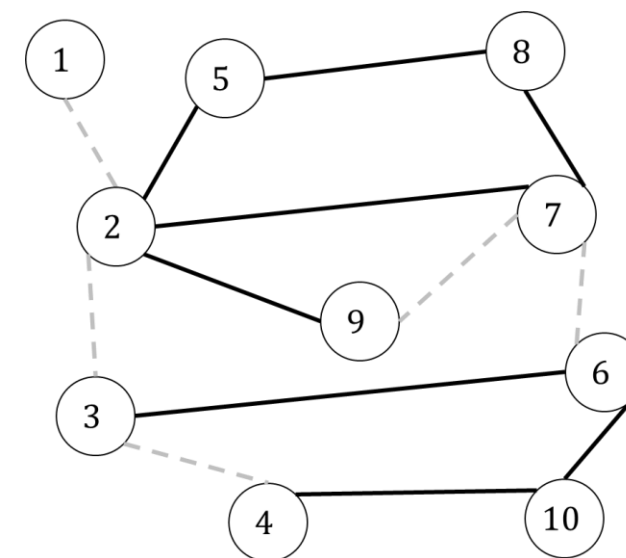
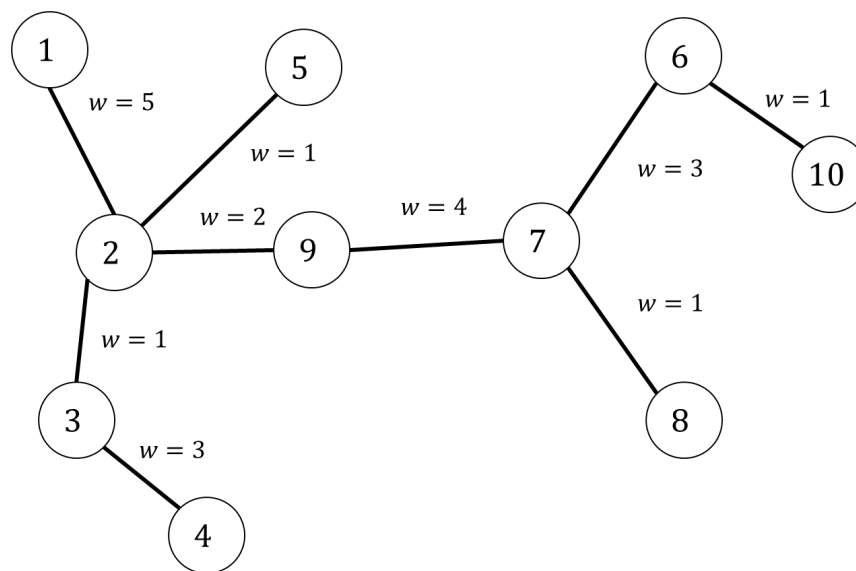
 :Liên kết cộng tác viết báo khoa học

 :Nghiên cứu viên

Author:ID
1
2
3
4
5
6
7
8
9
10



10 nghiên cứu viên



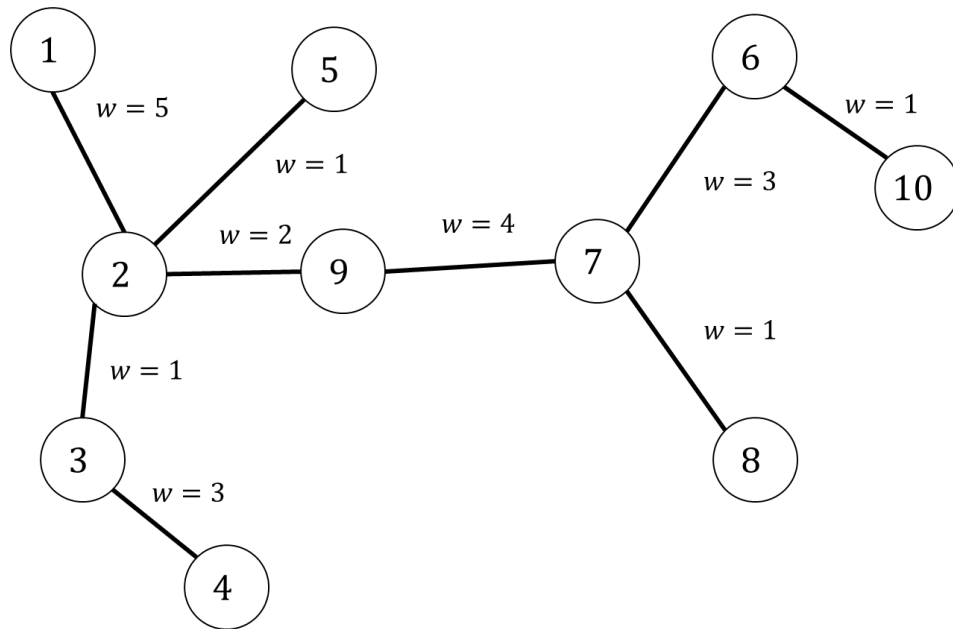
[2014-2015]

[2016-2017]

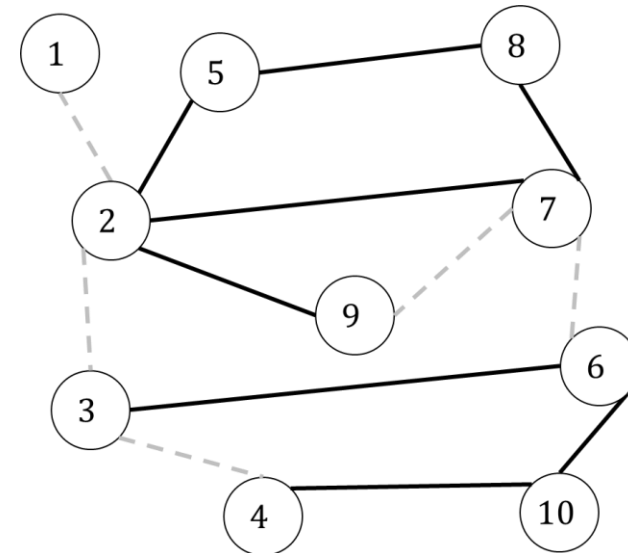
# 9. Ví dụ Node2vec

## 9.3. Yêu cầu:

Sử dụng Node2vec học cấu trúc mạng đồng tác giả 2014-2015 phục vụ tiên đoán liên kết 2016-2017.



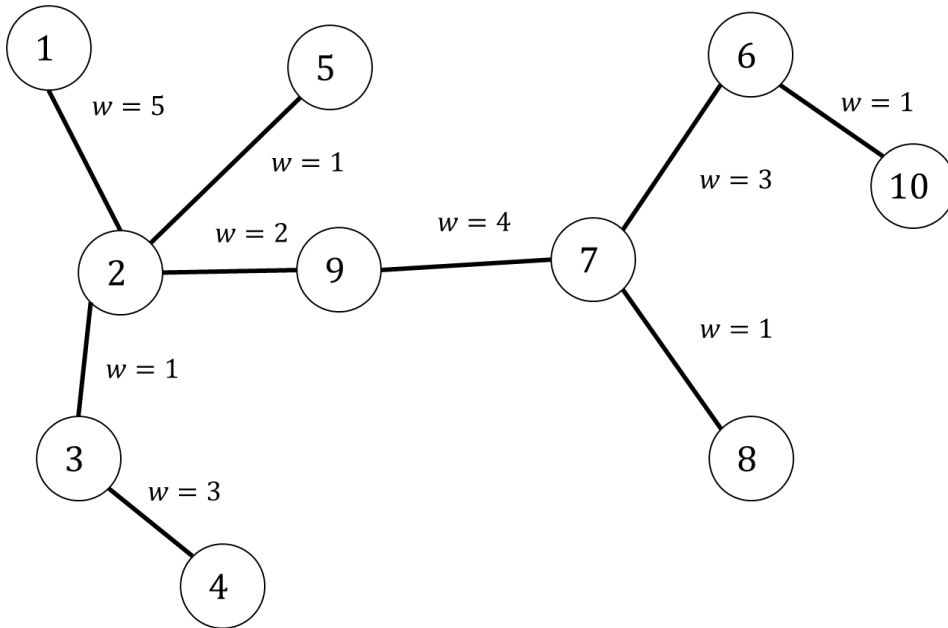
[2014-2015]



[2016-2017]

# 9. Ví dụ Node2vec

## 9.4. Bài toán:



[2014-2015]

### Đầu vào:

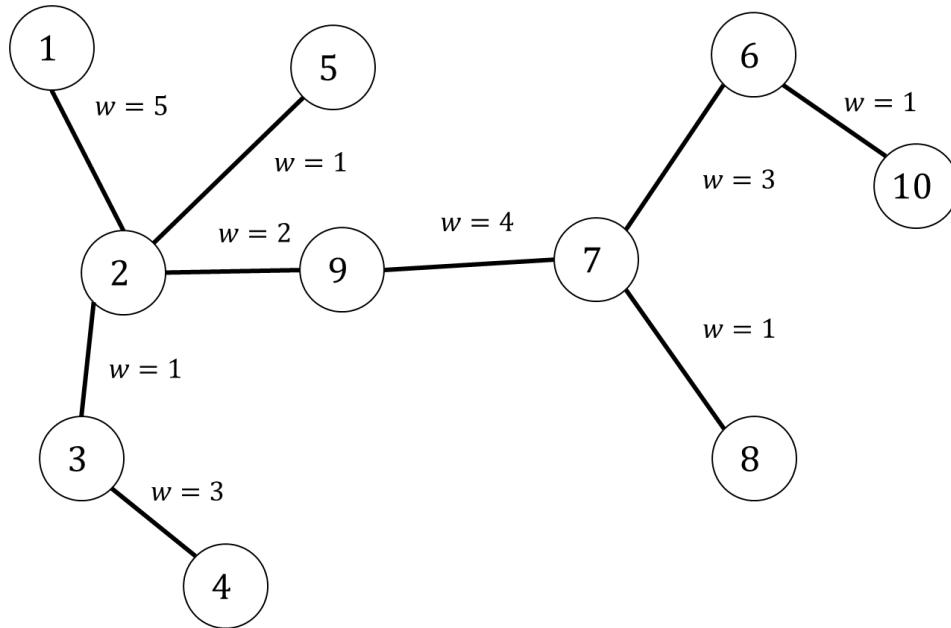
- Mạng đồng tác giả:  $CoNet = (U, Co_1)$
- Số chiều muốn biểu diễn:  $d = 5$
- Số bước từ mọi nút trong mạng:  $l=4$
- Số nghiên cứu viên tương đồng lấy từ tập bước:  $k = 2$  ( $l < k$ )
- Tham số xác định khả năng trở về của mỗi bước:  $p = 0.6$
- Tham số xác định khả năng bước tiếp tại mỗi bước:  $q = 1.3$

### Đầu ra:

$\forall u \in U$ , khuyến nghị  $n$  nghiên cứu viên có khả năng liên kết cộng tác với  $u$  trong 2016-2017.

## 9. Ví dụ Node2vec

### 9.5. Giải:



[2014-2015]

**Bước 1:** Tạo bảng random-walk cho nút theo tham số  $q$  và  $p$

**Bước 2:** Tìm cấu trúc liên kết của mạng theo bảng random-walk

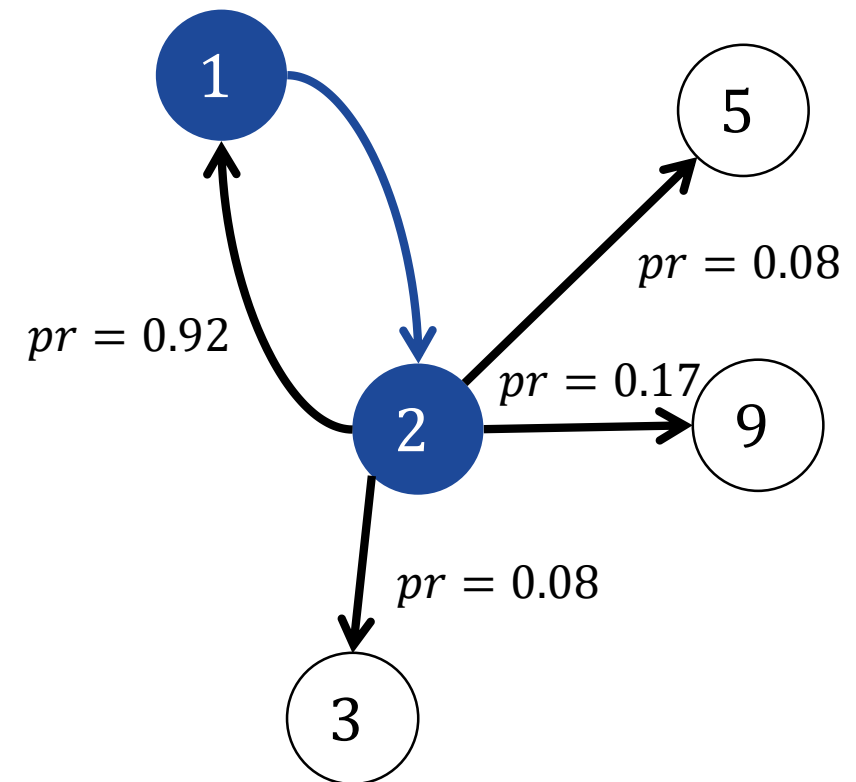
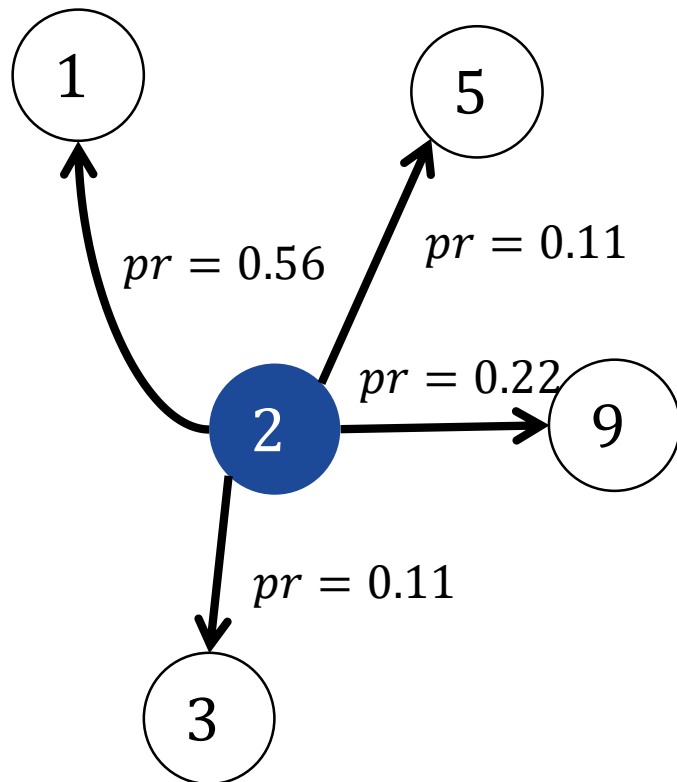
**Bước 3:** Tìm biểu diễn thỏa sự tương đồng giữa nút với tập lân cận của nút đó trong mạng

**Bước 4:** Tiên đoán liên kết và khuyến nghị.



**Bước 1:** Tạo bảng random-walk cho nút theo tham số  $q$  và  $p$

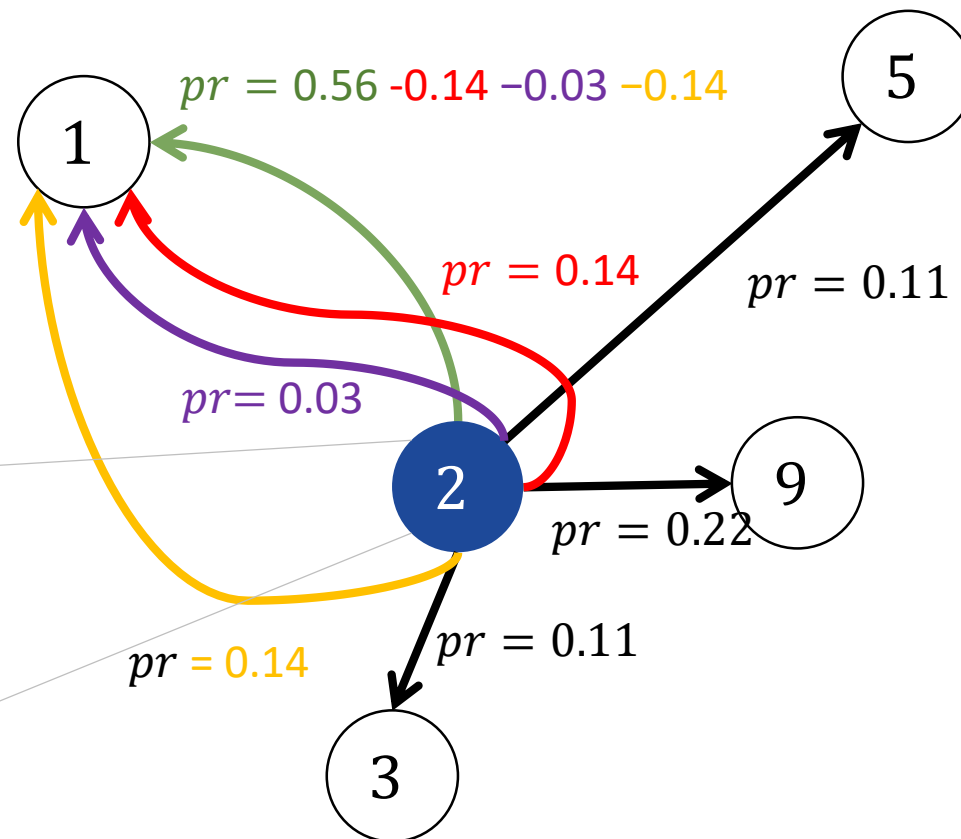
Probability transition



**Bước 1:** Tạo bảng random-walk cho nút theo tham số  $q$  và  $p$

k	Probability transition	
1	$u_1$ 0.25	
2	$u_9$ 0.22	$u_1$ 0.03
3	$u_3$ 0.11	$u_1$ 0.14
4	$u_5$ 0.11	$u_1$ 0.14

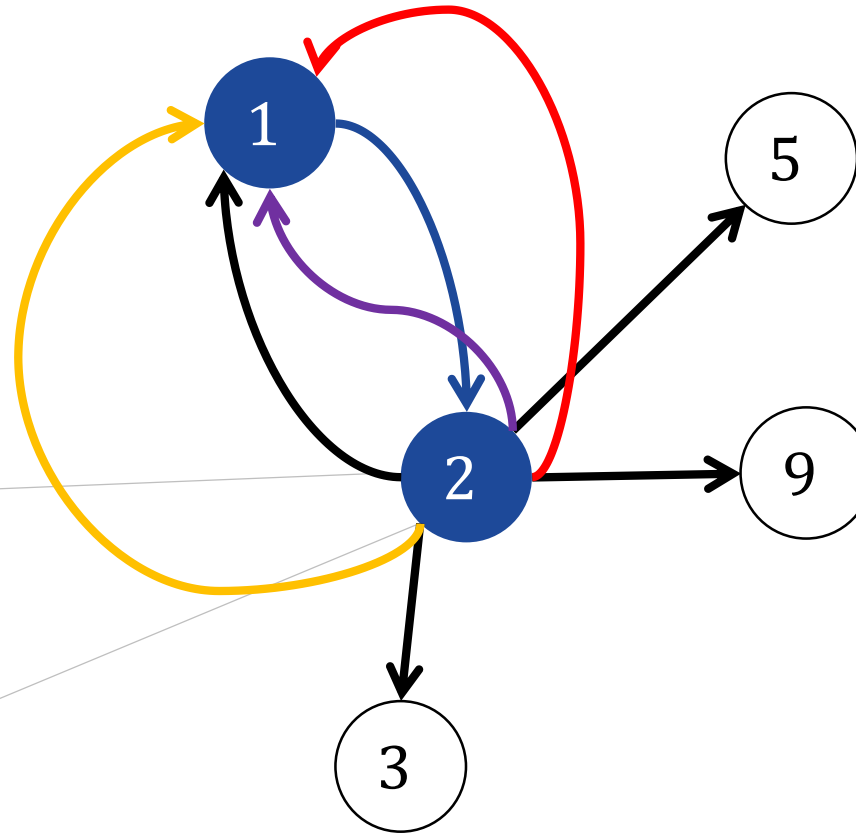
k	Probability walk	
1	$u_1$ 100%	
2	$u_9$ 44%	$u_1$ 100%
3	$u_3$ 88%	$u_1$ 100%
4	$u_5$ 44%	$u_1$ 100%



**Bước 1:** Tạo bảng random-walk cho nút theo tham số  $q$  và  $p$

k	Probability transition	
1	<div></div>	
2		<div></div>
3		<div></div>
4		<div></div>

k	Probability walk	
1	<div></div>	
2		<div></div>
3		<div></div>
4		<div></div>



**Bước 2:** Tìm cấu trúc liên kết của mạng  $Steps = [Step_l(u)]$ ,  $\forall u \in U$  theo bảng random-walk

Với  $u \in U$ ,  $Step(u) = \mathbf{BiasedRandomWalk}(u)$ :

$Step(u) = \{u\}$

**while**( $|Step(u)| < l$ ):

**if**( $|Step(u)| == 1$ ):

        neighbor = *RandomWalk*(u)

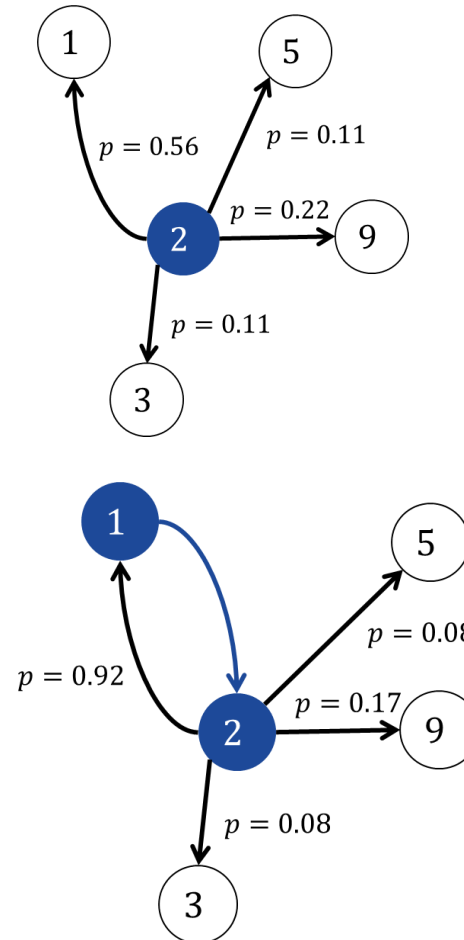
**else**:

        last =  $Step(u)[-1]$

        edge = ( $Step(u)[-2]$ , last)

        neighbor = *RandomWalk*(last, edge)

$Step(u) = \{u, neighbor\}$



**Bước 2:** Tìm cấu trúc liên kết của mạng  $Steps=[Step_l(u)]$ ,  $\forall u \in U$  theo bảng random-walk

**RandomWalk(u)**

RandomWalk(last, edge) *Tương tự*

k	Probability walk	
1	$u_1$ 100%	
2	$u_9$ 44%	$u_1$ 100%
3	$u_3$ 88%	$u_1$ 100%
4	$u_5$ 44%	$u_1$ 100%

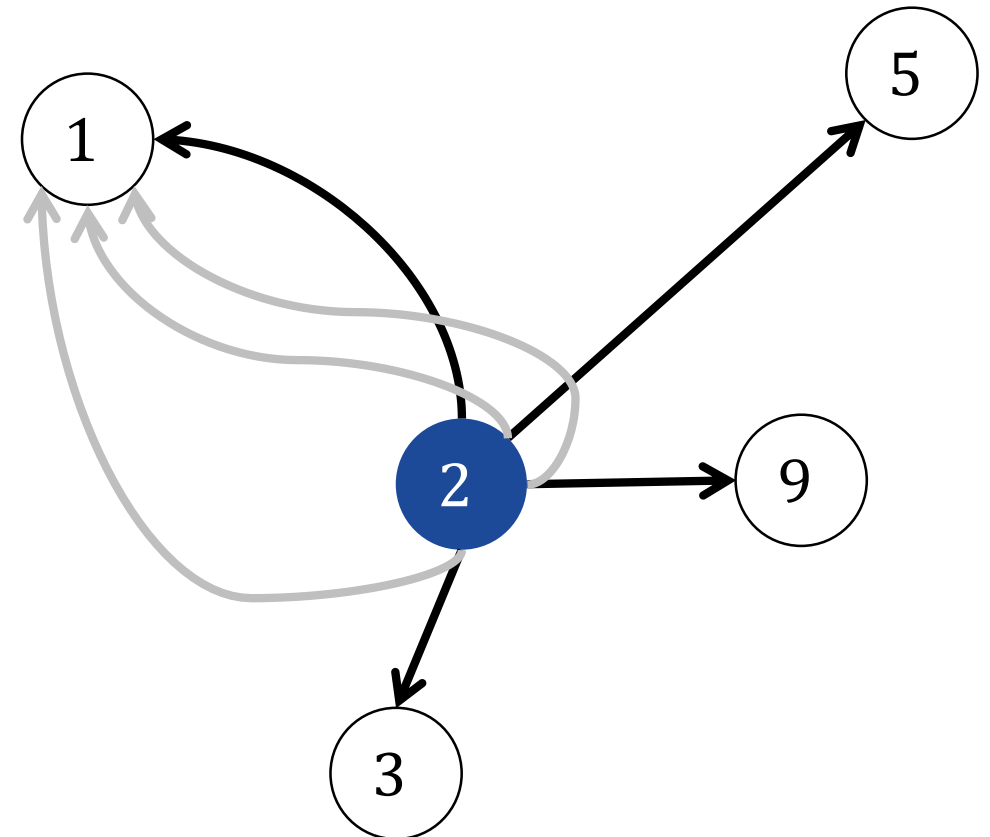


k= 3



Randomly pick = 0.35 < 0.88

→ Stay at  $u_9$



**Bước 3:** Tìm biểu diễn thỏa sự tương đồng giữa nút với tập lân cận của nút đó trong mạng

```
 $f = \text{Word2vec}(\text{corpus} = \text{Steps},$   
     $\text{dimension} = d = 5,$   
     $\text{window}_{\text{size}} = k = 2,$   
     $\text{skip\_gram} = \text{true} // \text{Dùng mô hình Skip-gram}$   
     $)$ 
```

**Giả định kết quả bước 2:**  
***Steps*** =  $[ \dots, \{u_2, u_9, u_7, u_6\}, \dots ]$

$f: U \rightarrow \mathbb{R}^d$  là ma trận gồm các vector ánh xạ những nghiên cứu viên sang không gian vector.

**Bước 3:** Tìm biểu diễn thỏa sự tương đồng giữa nút với tập lân cận của nút đó trong mạng

### 3.1 Tìm nghiên cứu viên lân cận

#1	"2"	"9"	"7"	"6"
	X <sub>k</sub>	Y(c=1)	Y(c=2)	
#2	"2"	"9"	"7"	"6"
	Y(c=1)	X <sub>k</sub>	Y(c=2)	Y(c=3)
#3	"2"	"9"	"7"	"6"
	Y(c=1)	X <sub>k</sub>	Y(c=2)	Y(c=3)
#4	"2"	"9"	"7"	"6"
	Y(c=1)	X <sub>k</sub>	Y(c=2)	Y(c=3)

**Bước 3:** Tìm biểu diễn thỏa sự tương đồng giữa nút với tập lân cận của nút đó trong mạng

### 3.2 Biểu diễn one-hot vector

0	0	0
1	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	1
0	0	0
0	1	0
0	0	0
<b>X<sub>k</sub></b>	<b>Y(c=1)</b>	<b>Y(c=2)</b>



### 3.3 Sử dụng Skip-gram tìm biểu diễn $f$ thỏa sự tương đồng với nút lân cận

3.3.1  $f: W \rightarrow \mathbb{R}^d$ .

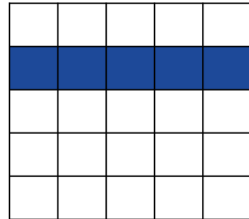
3.3.2  $\max_f \sum_{u \in V} \log(\Pr(N_{\mathbf{s}}(u)|f(u)))$

Input layer

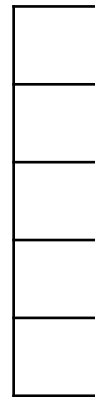


1 x 10

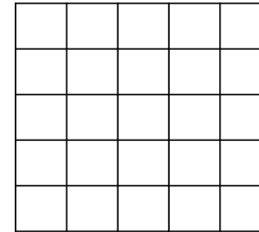
Hidden layer



10 x 5

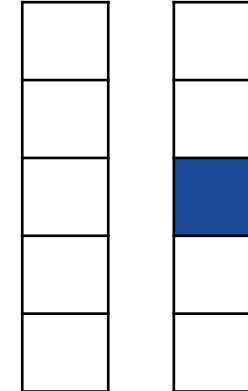


1 x 5

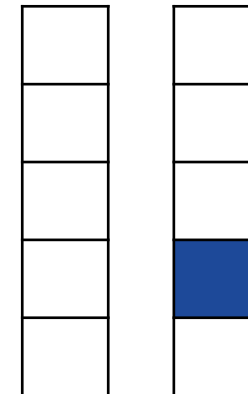


5 x 10

Output layer

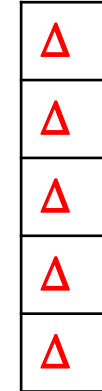


1 x 10 1 x 10

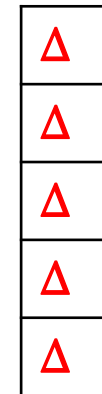


1 x 10

dif

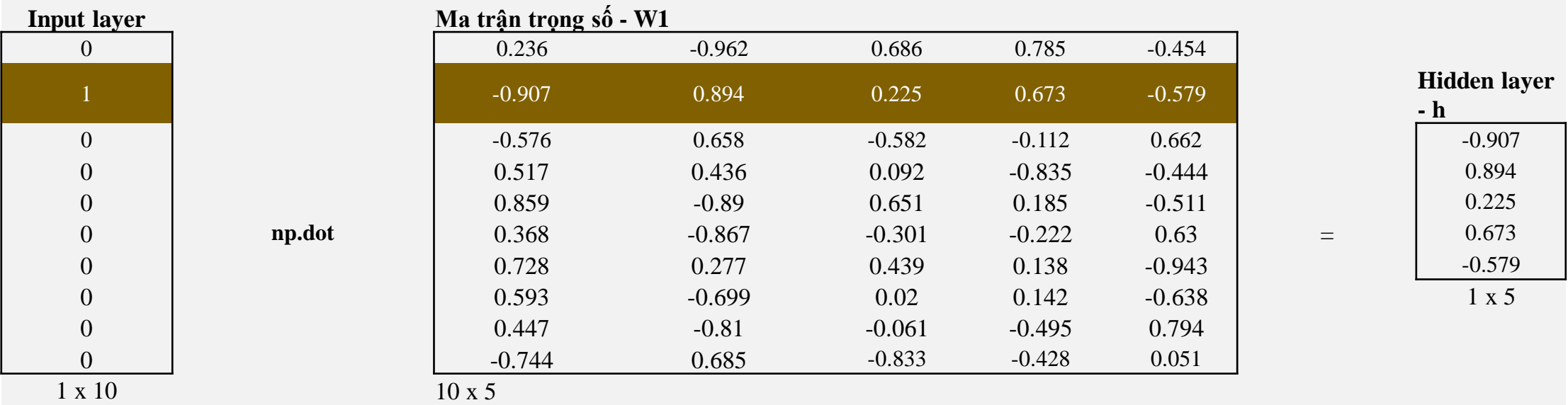


1 x 10



### 3.3 Sử dụng Skip-gram tìm biểu diễn $f$ thỏa sự tương đồng với nút lân cận

#### 3.3.1 $f: W \rightarrow \mathbb{R}^d$ .





3.3.2  $\max_f \sum_{u \in V} \log(\text{Pr}(N_{\textcolor{red}{s}}(u)|f(u)))$

Hidden layer - h

-0.907
0.894
0.225
0.673
-0.579

1 x 5

Ma trận  
trọng số - W2

0.859	-0.89	0.651	0.185	-0.511	-0.456	0.377	-0.274	0.182	-0.237
0.368	-0.867	-0.301	-0.222	0.63	0.808	0.088	-0.902	-0.45	-0.408
0.728	0.277	0.439	0.138	-0.943	-0.409	0.687	-0.215	-0.807	0.612
0.593	-0.699	0.02	0.142	-0.638	-0.633	0.344	0.868	0.913	0.429
0.447	-0.81	-0.061	-0.495	0.794	-0.064	-0.817	-0.408	-0.286	0.149

5 x 10

np.dot

=

Output layer -  
no softmax

-0.146
0.093
-0.712
0.046
-0.07
0.65
0.59
0.21
0.03
0.19

1 x 10

Output layer -  
softmax

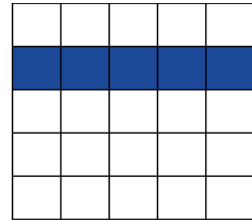
0.07
0.094
0.042
0.09
0.08
0.16
0.15
0.1
0.088
0.1

1 x 10

**Input layer**

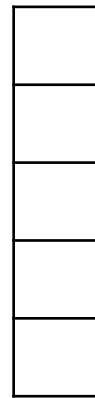


**1 x 10**

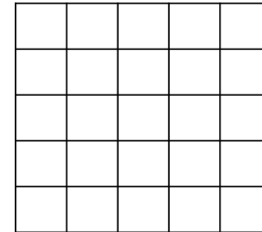


**10 x 5**

**Hidden layer**



**1 x 5**



**5 x 10**

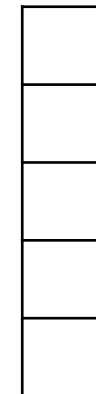
**Output layer**



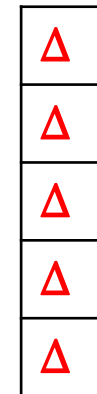
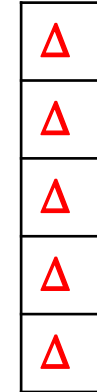
**1 x 10**



**1 x 10**



**dif**



**1 x 10**



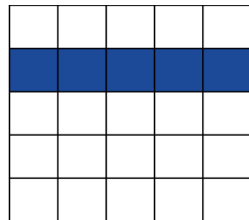
<b>Hidden layer - h</b>		<b>Ma trận trọng số - W2</b>		<b>Output layer - no softmax</b>		<b>Output layer - softmax</b>																																																																											
<table><tr><td>-0.907</td></tr><tr><td>0.894</td></tr><tr><td>0.225</td></tr><tr><td>0.673</td></tr><tr><td>-0.579</td></tr></table>	-0.907	0.894	0.225	0.673	-0.579	<b>np.dot</b>	<table><tr><td>0.859</td><td>-0.89</td><td>0.651</td><td>0.185</td><td>-0.511</td><td>-0.456</td><td>0.377</td><td>-0.274</td><td>0.182</td><td>-0.237</td></tr><tr><td>0.368</td><td>-0.867</td><td>-0.301</td><td>-0.222</td><td>0.63</td><td>0.808</td><td>0.088</td><td>-0.902</td><td>-0.45</td><td>-0.408</td></tr><tr><td>0.728</td><td>0.277</td><td>0.439</td><td>0.138</td><td>-0.943</td><td>-0.409</td><td>0.687</td><td>-0.215</td><td>-0.807</td><td>0.612</td></tr><tr><td>0.593</td><td>-0.699</td><td>0.02</td><td>0.142</td><td>-0.638</td><td>-0.633</td><td>0.344</td><td>0.868</td><td>0.913</td><td>0.429</td></tr><tr><td>0.447</td><td>-0.81</td><td>-0.061</td><td>-0.495</td><td>0.794</td><td>-0.064</td><td>-0.817</td><td>-0.408</td><td>-0.286</td><td>0.149</td></tr></table>	0.859	-0.89	0.651	0.185	-0.511	-0.456	0.377	-0.274	0.182	-0.237	0.368	-0.867	-0.301	-0.222	0.63	0.808	0.088	-0.902	-0.45	-0.408	0.728	0.277	0.439	0.138	-0.943	-0.409	0.687	-0.215	-0.807	0.612	0.593	-0.699	0.02	0.142	-0.638	-0.633	0.344	0.868	0.913	0.429	0.447	-0.81	-0.061	-0.495	0.794	-0.064	-0.817	-0.408	-0.286	0.149	<b>=</b>	<table><tr><td>-0.146</td></tr><tr><td>0.093</td></tr><tr><td>-0.712</td></tr><tr><td>0.046</td></tr><tr><td>-0.07</td></tr><tr><td>0.65</td></tr><tr><td>0.59</td></tr><tr><td>0.21</td></tr><tr><td>0.03</td></tr><tr><td>0.19</td></tr></table>	-0.146	0.093	-0.712	0.046	-0.07	0.65	0.59	0.21	0.03	0.19		<table><tr><td>0.07</td></tr><tr><td>0.094</td></tr><tr><td>0.042</td></tr><tr><td>0.09</td></tr><tr><td>0.08</td></tr><tr><td>0.16</td></tr><tr><td>0.15</td></tr><tr><td>0.1</td></tr><tr><td>0.088</td></tr><tr><td>0.1</td></tr></table>	0.07	0.094	0.042	0.09	0.08	0.16	0.15	0.1	0.088	0.1
-0.907																																																																																	
0.894																																																																																	
0.225																																																																																	
0.673																																																																																	
-0.579																																																																																	
0.859	-0.89	0.651	0.185	-0.511	-0.456	0.377	-0.274	0.182	-0.237																																																																								
0.368	-0.867	-0.301	-0.222	0.63	0.808	0.088	-0.902	-0.45	-0.408																																																																								
0.728	0.277	0.439	0.138	-0.943	-0.409	0.687	-0.215	-0.807	0.612																																																																								
0.593	-0.699	0.02	0.142	-0.638	-0.633	0.344	0.868	0.913	0.429																																																																								
0.447	-0.81	-0.061	-0.495	0.794	-0.064	-0.817	-0.408	-0.286	0.149																																																																								
-0.146																																																																																	
0.093																																																																																	
-0.712																																																																																	
0.046																																																																																	
-0.07																																																																																	
0.65																																																																																	
0.59																																																																																	
0.21																																																																																	
0.03																																																																																	
0.19																																																																																	
0.07																																																																																	
0.094																																																																																	
0.042																																																																																	
0.09																																																																																	
0.08																																																																																	
0.16																																																																																	
0.15																																																																																	
0.1																																																																																	
0.088																																																																																	
0.1																																																																																	
1 x 5		5 x 10		1 x 10		1 x 10																																																																											

**Input layer**



**1 x 10**

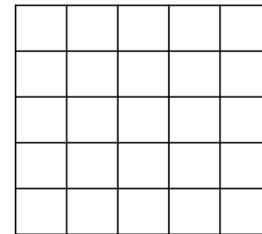
**Hidden layer**



**10 x 5**



**1 x 5**



**5 x 10**

**Output layer**

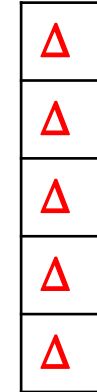


**1 x 10**



**1 x 10**

**dif**



**1 x 10**

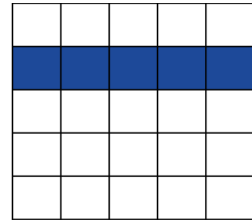


Output layer - softmax			Diff	Output layer - softmax			Diff
0.07	0	0.07		0.07	0	0.07	0.14
0.094	0	0.094		0.094	0	0.094	0.188
0.042	0	0.042		0.042	0	0.042	0.084
0.09	0	0.09		0.09	0	0.09	0.18
0.08	0	0.08		0.08	0	0.08	0.16
0.16	0	0.16		0.16	0	0.16	0.32
0.15	0	0.15		0.15	1	-0.85	-0.7
0.1	0	0.1		0.1	0	0.1	0.2
0.088	1	0.912		0.088	0	0.088	1
0.1	0	0.1		0.1	0	0.1	0.2

**Input layer**

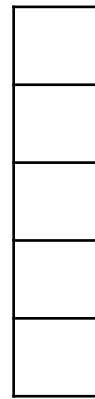


**1 x 10**

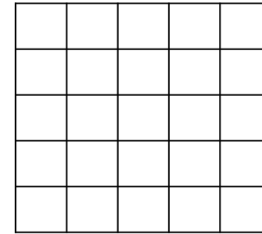


**10 x 5**

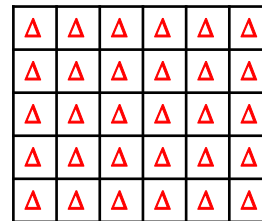
**Hidden layer**



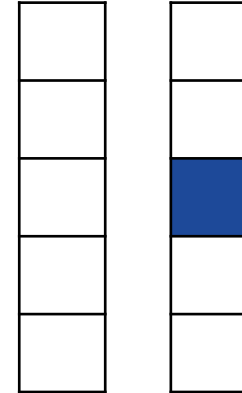
**1 x 5**



**5 x 10**



**Output layer**

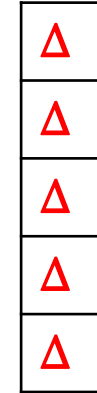


**1 x 10**



**1 x 10**

**dif**



**1 x 10**





**Hidden layer**

-0.907
0.894
0.225
0.673
-0.579

5 x 1

**np.outer**

0.14
0.188
0.084
0.18
0.16
0.32
-0.7
0.2
1
0.2

10 x 1

=

**Delta cho W2**

-0.12698	-0.16326	-0.07256	-0.16326	-0.14512	-0.29024	0.6349	-0.1814	-0.907	-0.1814
0.12516	0.16092	0.07152	0.16092	0.14304	0.28608	-0.6258	0.1788	0.894	0.1788
0.0315	0.0405	0.018	0.0405	0.036	0.072	-0.1575	0.045	0.225	0.045
0.09422	0.12114	0.05384	0.12114	0.10768	0.21536	-0.4711	0.1346	0.673	0.1346
-0.08106	-0.10422	-0.04632	-0.10422	-0.09264	-0.18528	0.4053	-0.1158	-0.579	-0.1158

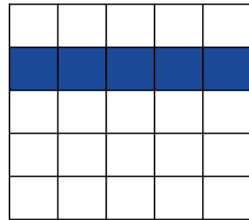
5 x 10

**Input layer**

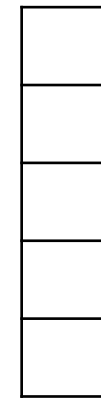
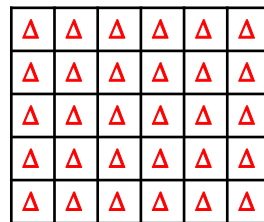


**1 x 10**

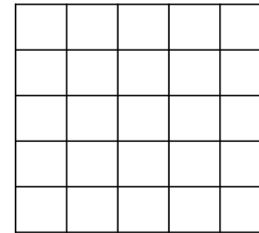
**Hidden layer**



**10 x 5**

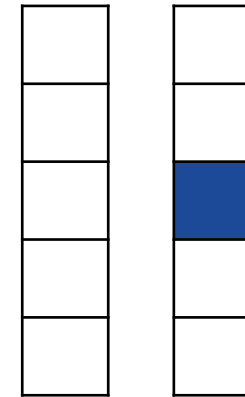


**1 x 5**

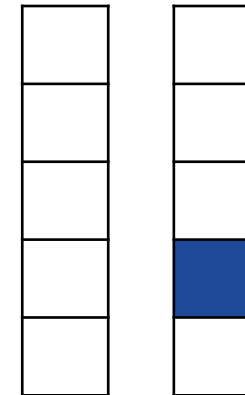


**5 x 10**

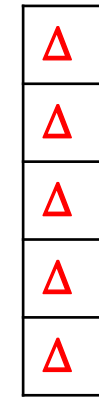
**Output layer**



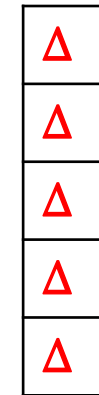
**1 x 10**   **1 x 10**



**dif**



**1 x 10**





W2

0.859	-0.89	0.651	0.185	-0.511	-0.456	0.377	-0.274	0.182	-0.237
0.368	-0.867	-0.301	-0.222	0.63	0.808	0.088	-0.902	-0.45	-0.408
0.728	0.277	0.439	0.138	-0.943	-0.409	0.687	-0.215	-0.807	0.612
0.593	-0.699	0.02	0.142	-0.638	-0.633	0.344	0.868	0.913	0.429
0.447	-0.81	-0.061	-0.495	0.794	-0.064	-0.817	-0.408	-0.286	0.149

5 x 10

np.dot

error

0.14
0.188
0.084
0.18
0.16
0.32
-0.7
0.2
1
0.2

10 x 1

=

np.dot(W2,EI)

-0.37
-0.59
-1.27
0.6
0.16

5 x 1

Input layer

0
1
0
0
0
0
0
0
0
0

1 x 10

np.outer

np.dot(W2,EI)

-0.37
-0.59
-1.27
0.6
0.16

5 x 1

=

Delta cho W1

0	0	0	0	0
-0.37	-0.59	-1.27	0.6	0.16
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

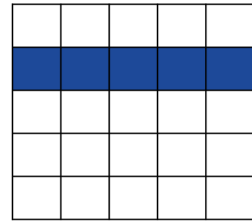
10 x 5

**Input layer**

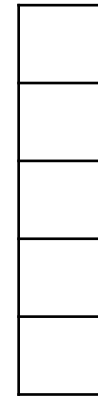
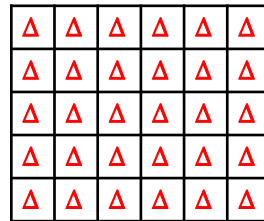


**1 x 10**

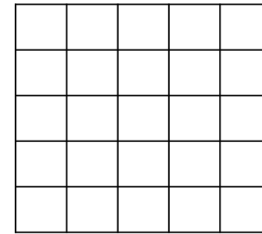
**Hidden layer**



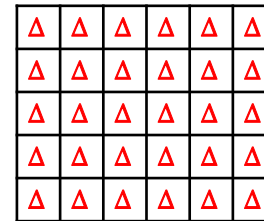
**10 x 5**



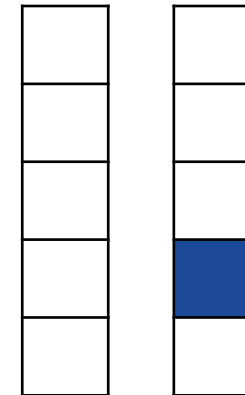
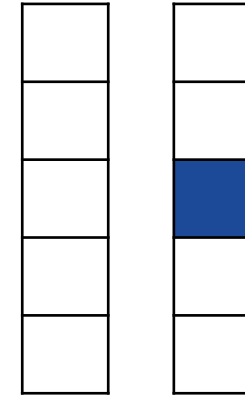
**1 x 5**



**5 x 10**



**Output layer**



**1 x 10**

**1 x 10**

**dif**

**1 x 10**



**Ma trận trọng số - W1**

0.236	-0.962	0.686	0.785	-0.454
-0.907	0.894	0.225	0.673	-0.579
-0.576	0.658	-0.582	-0.112	0.662
0.517	0.436	0.092	-0.835	-0.444
0.859	-0.89	0.651	0.185	-0.511
0.368	-0.867	-0.301	-0.222	0.63
0.728	0.277	0.439	0.138	-0.943
0.593	-0.699	0.02	0.142	-0.638
0.447	-0.81	-0.061	-0.495	0.794
-0.744	0.685	-0.833	-0.428	0.051

10 x 5

**Delta cho W1**

0	0	0	0	0
-0.37	-0.59	-1.27	0.6	0.16
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

10 x 5

- 

<b>Learning Rate</b>
0.1

 x

**Ma trận trọng số cập nhật - W1**

0.236	-0.962	0.686	0.785	-0.454
-0.87	0.953	0.352	0.613	-0.595
-0.576	0.658	-0.582	-0.112	0.662
0.517	0.436	0.092	-0.835	-0.444
0.859	-0.89	0.651	0.185	-0.511
0.368	-0.867	-0.301	-0.222	0.63
0.728	0.277	0.439	0.138	-0.943
0.593	-0.699	0.02	0.142	-0.638
0.447	-0.81	-0.061	-0.495	0.794
-0.744	0.685	-0.833	-0.428	0.051

10 x 5

**Mã trận trọng số - W2**

0.859	-0.89	0.651	0.185	-0.511	-0.456	0.377	-0.274	0.182	-0.237
0.368	-0.867	-0.301	-0.222	0.63	0.808	0.088	-0.902	-0.45	-0.408
0.728	0.277	0.439	0.138	-0.943	-0.409	0.687	-0.215	-0.807	0.612
0.593	-0.699	0.02	0.142	-0.638	-0.633	0.344	0.868	0.913	0.429
0.447	-0.81	-0.061	-0.495	0.794	-0.064	-0.817	-0.408	-0.286	0.149

5 x 10

-

Learning Rate

0.1

x

**Delta cho W2**

-0.12698	-0.16326	-0.07256	-0.16326	-0.14512	-0.2902	0.6349	-0.1814	-0.907	-0.1814	=
0.12516	0.16092	0.07152	0.16092	0.14304	0.2861	-0.6258	0.1788	0.894	0.1788	
0.0315	0.0405	0.018	0.0405	0.036	0.072	-0.1575	0.045	0.225	0.045	
0.09422	0.12114	0.05384	0.12114	0.10768	0.2154	-0.4711	0.1346	0.673	0.1346	
-0.08106	-0.10422	-0.04632	-0.10422	-0.09264	-0.1853	0.4053	-0.1158	-0.579	-0.1158	

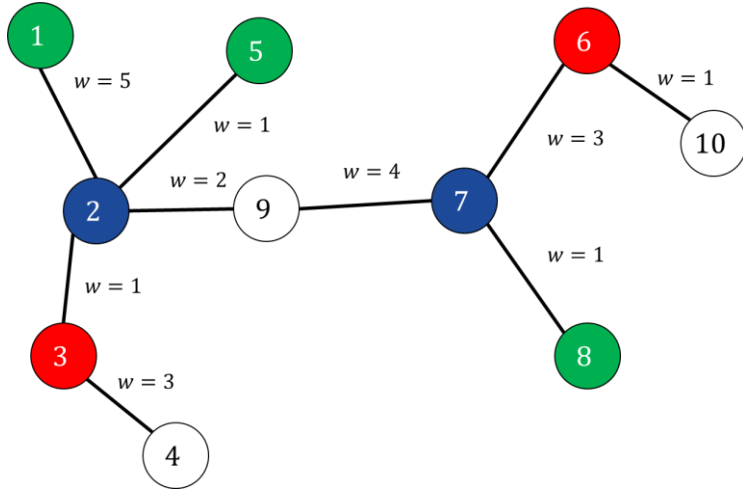
5 x 10

**Mã trận trọng số cập nhật W2**

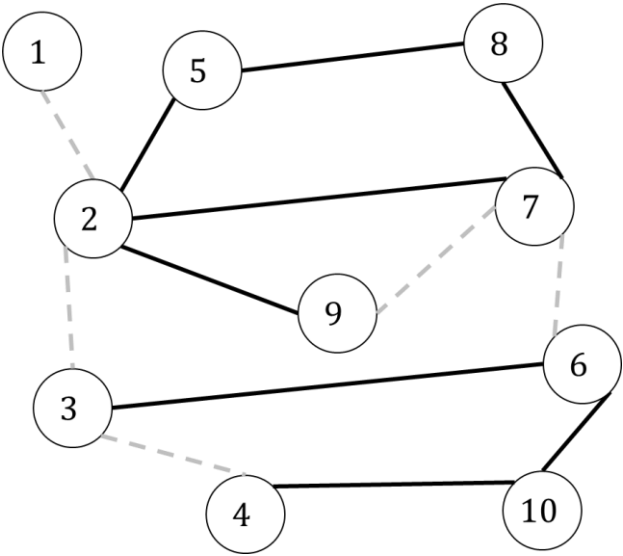
0.8717	-0.8737	0.6583	0.2013	-0.4965	-0.427	0.3135	-0.2559	0.2727	-0.2189
0.35548	-0.8831	-0.3082	-0.2381	0.6157	0.7794	0.1506	-0.9199	-0.5394	-0.4259
0.72485	0.273	0.4372	0.134	-0.9466	-0.4162	0.7028	-0.2195	-0.8295	0.6075
0.58358	-0.7111	0.0146	0.1299	-0.6488	-0.6545	0.3911	0.8545	0.8457	0.4155
0.45511	-0.7996	-0.0564	-0.4846	0.8033	-0.0455	-0.8575	-0.3964	-0.2281	0.1606

**Bước 4:** Tiên đoán liên kết và khuyến nghị.

$u_2$	-0.87	0.953	0.352	0.613	-0.595
$u_1$					



2014-2015 (Tiên đoán )



2016-2017