Metapath2vec

Antonio Longa[1,2]

MobS[1] Lab, Fondazione Bruno Kessler,Trento, Italy
SML[2] Lab, University of Trento, Italy

DONG, Yuxiao; CHAWLA, Nitesh V.; SWAMI, Ananthram. metapath2vec: Scalable representation learning for heterogeneous networks. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017. p. 135-144.

TABLE OF CONTENTS

01 Metapath

02 Metapath random walks

03 The model

04 **Training**
(my laptop not powerful enough😊)

05 Load a pre trained model

06

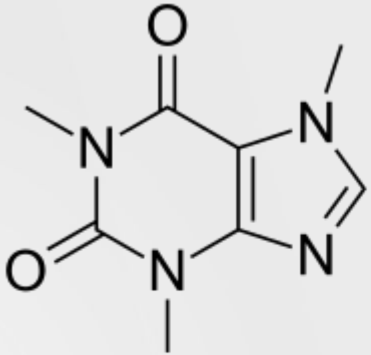
01 Metapath

So far we sow some **homogeneous** type of graphs:

01 Metapath

So far we sow some **homogeneous** type of graphs:

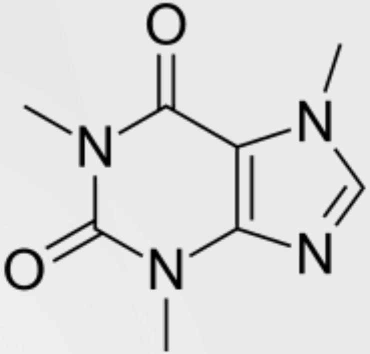
- **Moleculas**
- CiteSeer
- Cora

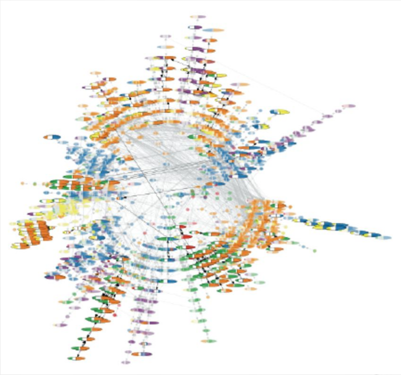


01 Metapath

So far we sow some **homogeneous** type of graphs:

- Moleculas
- **CiteSeer**
- Cora

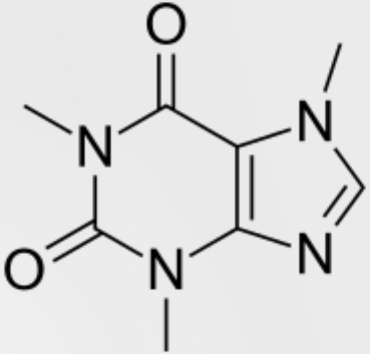


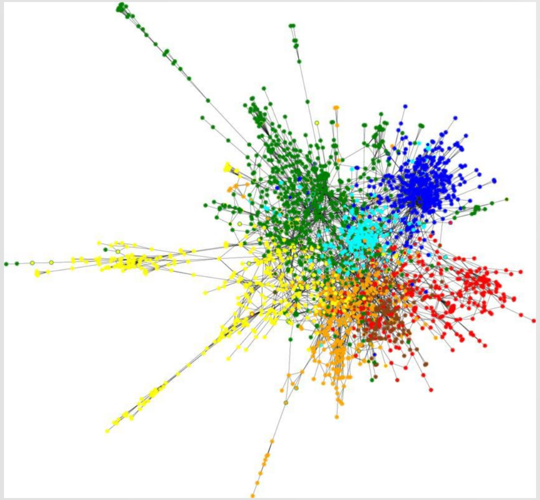


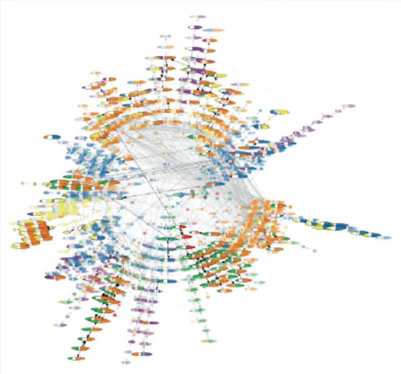
01 Metapath

So far we sow some **homogeneous** type of graphs:

- Moleculas
- CiteSeer
- **Cora**







01 Metapath

What are **heterogeneous** graph?

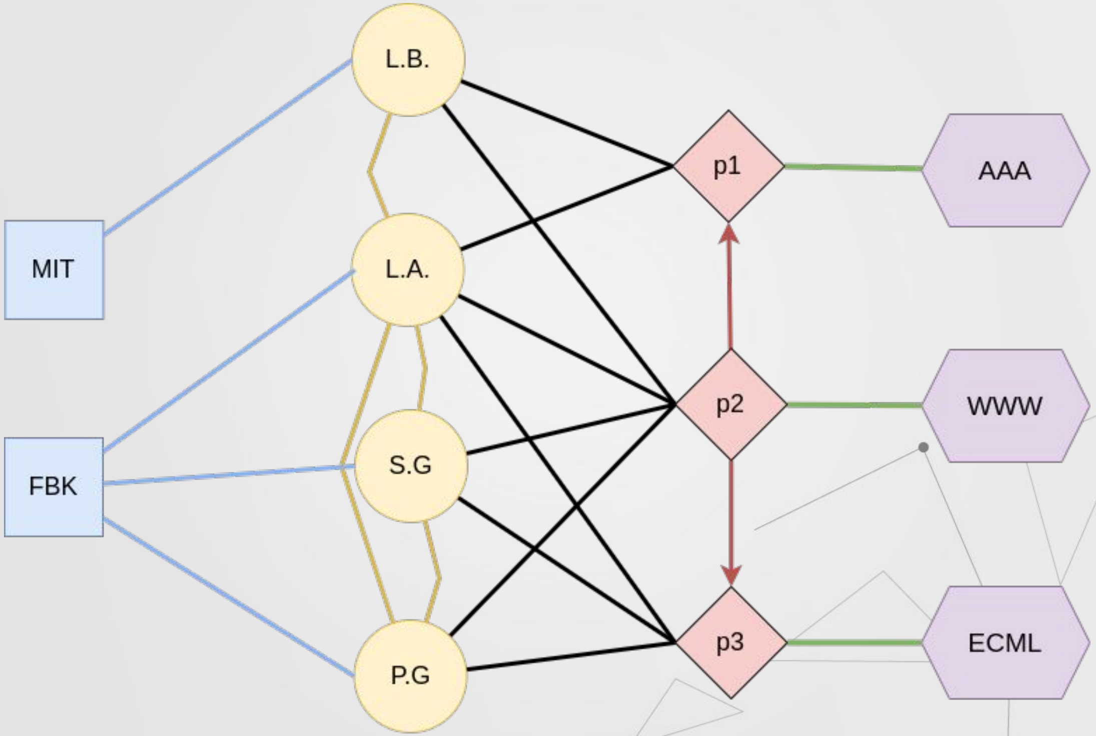
01 Metapath

What are **heterogeneous** graph?

An heterogeneous graph is a graph in witch nodes and edges have differents types.

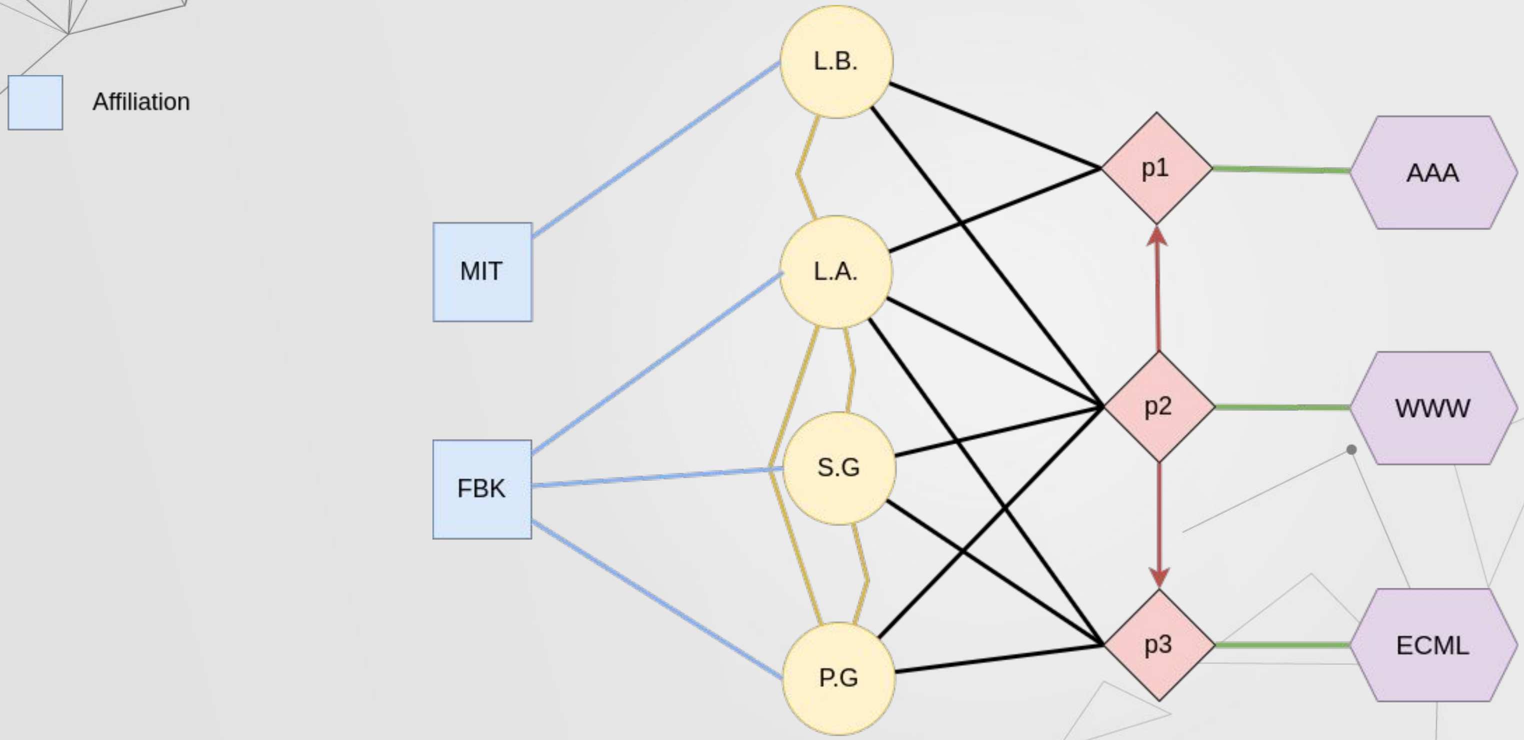
01 Metapath

For instance:



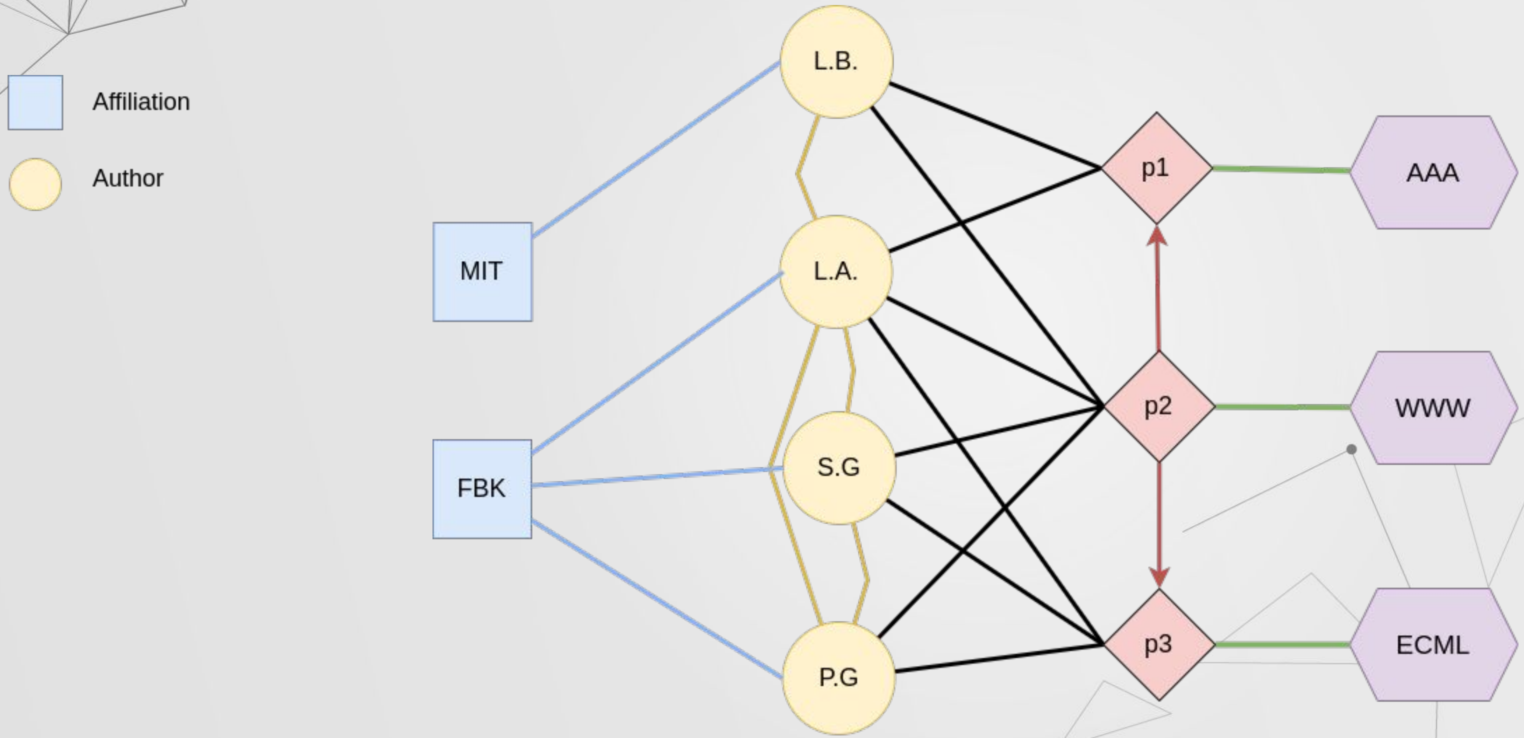
01 Metapath

For instance:



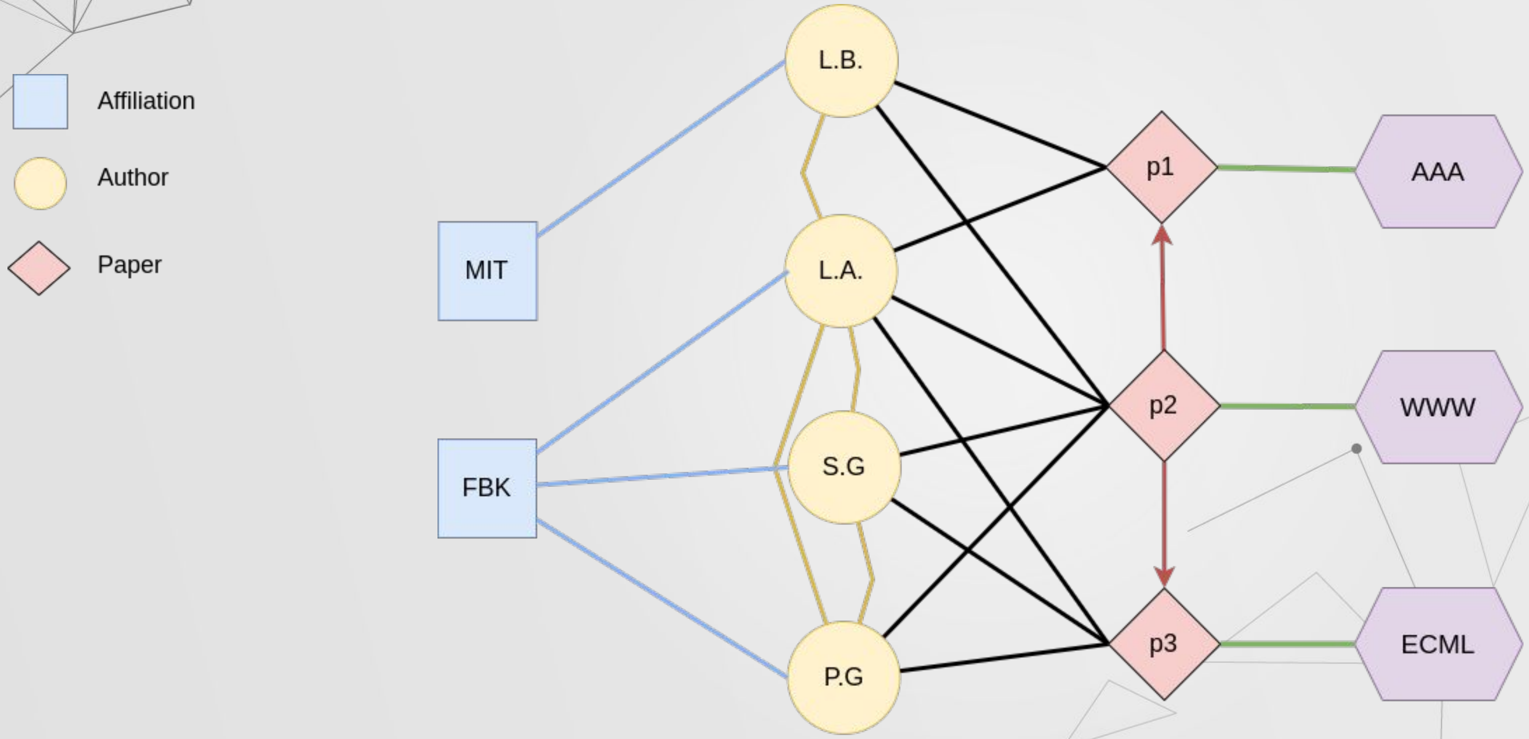
01 Metapath

For instance:



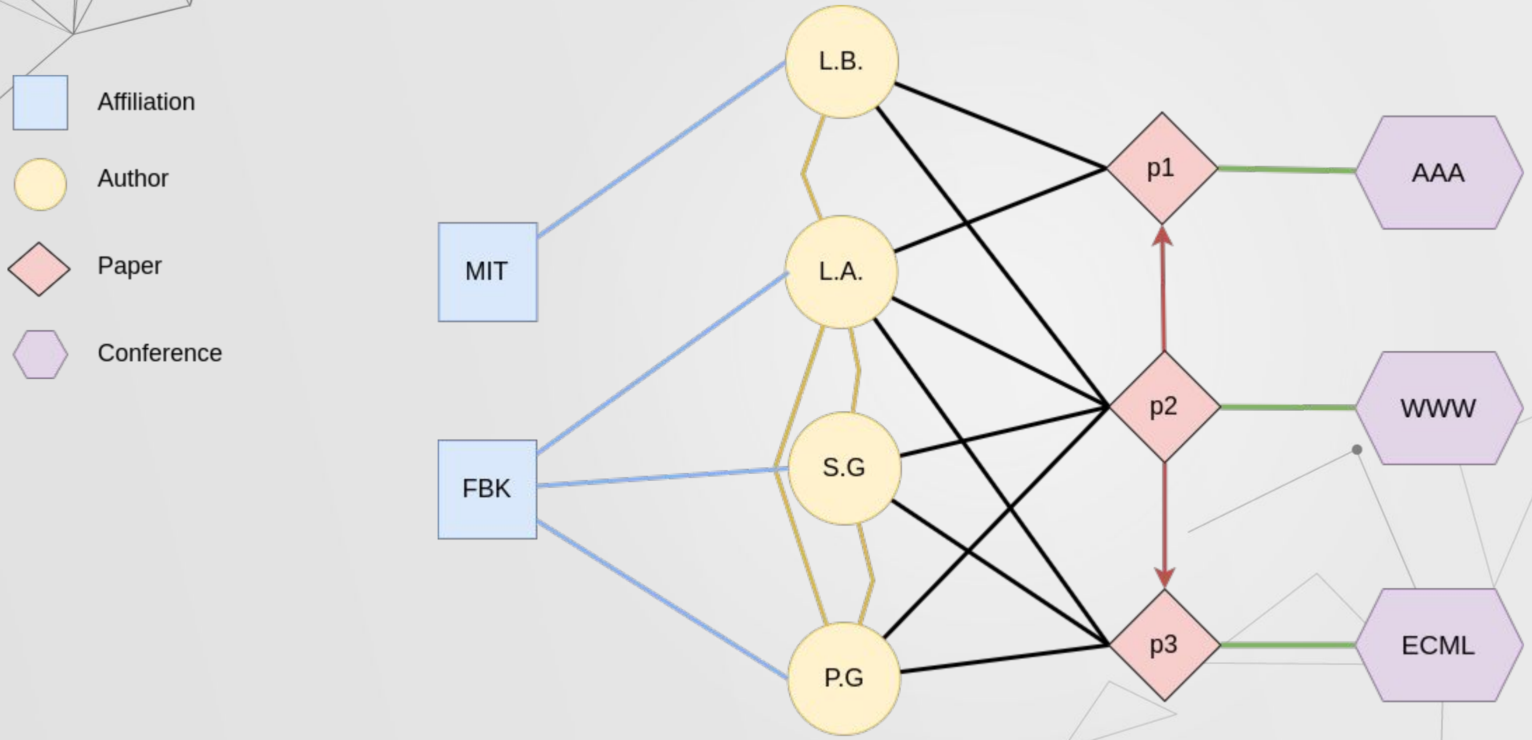
01 Metapath

For instance:



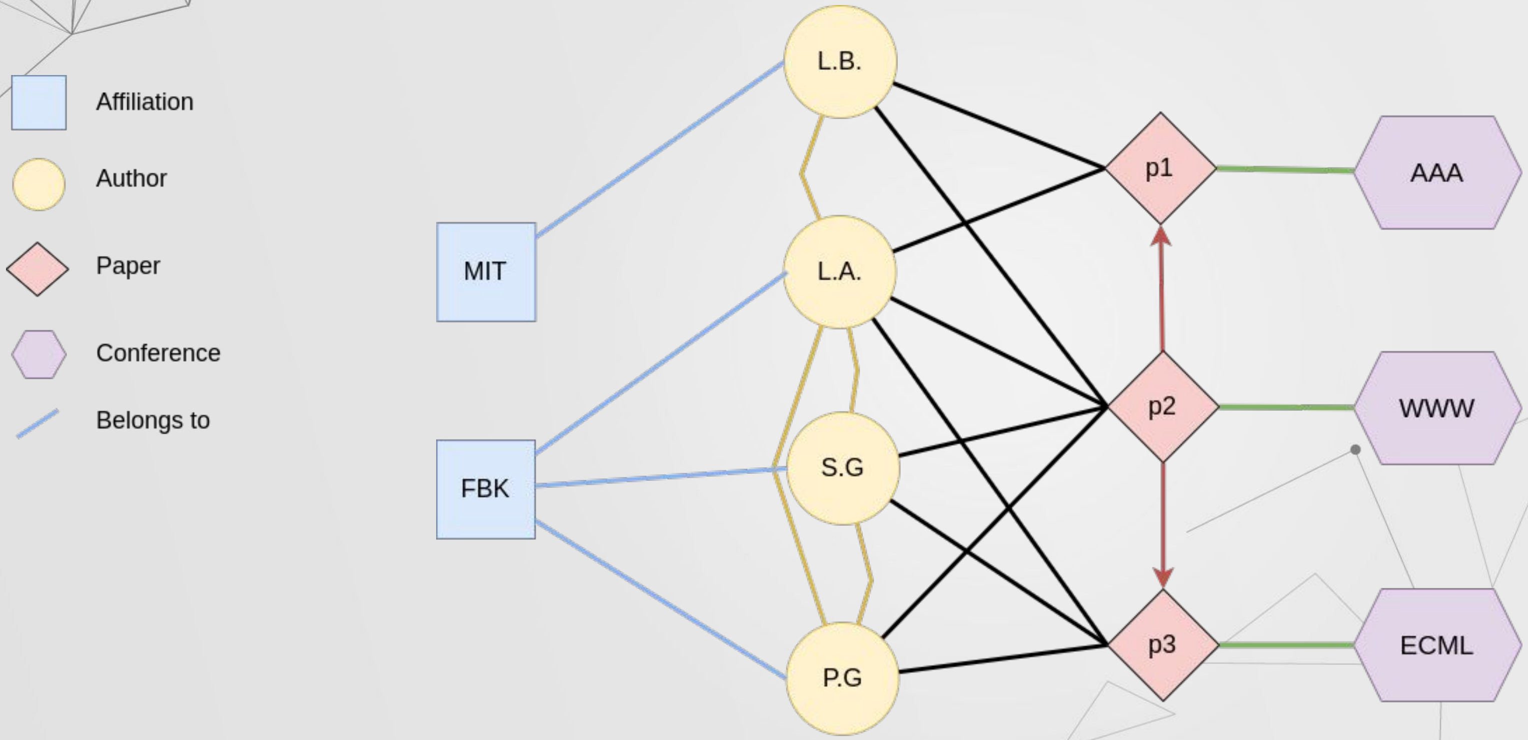
01 Metapath

For instance:



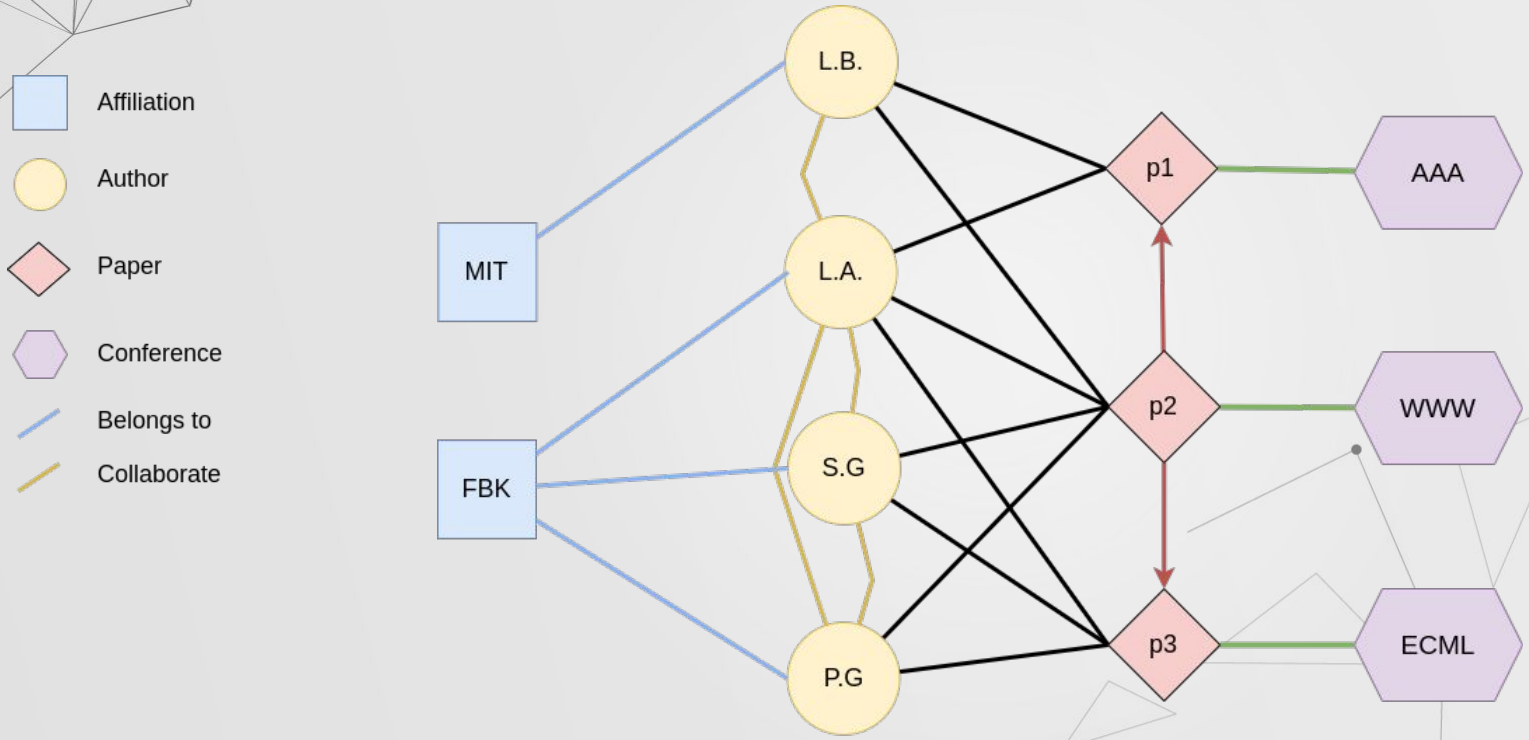
01 Metapath

For instance:



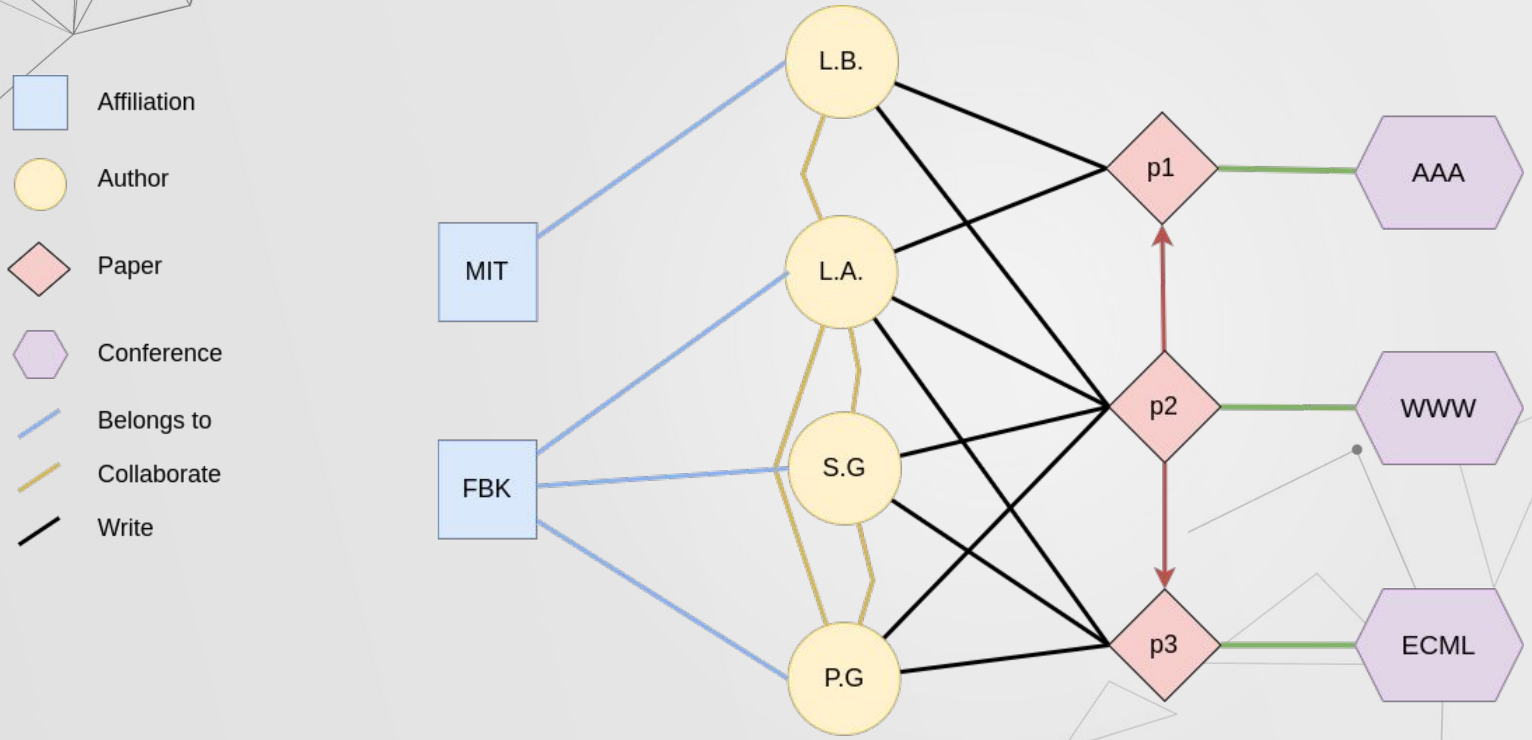
01 Metapath

For instance:



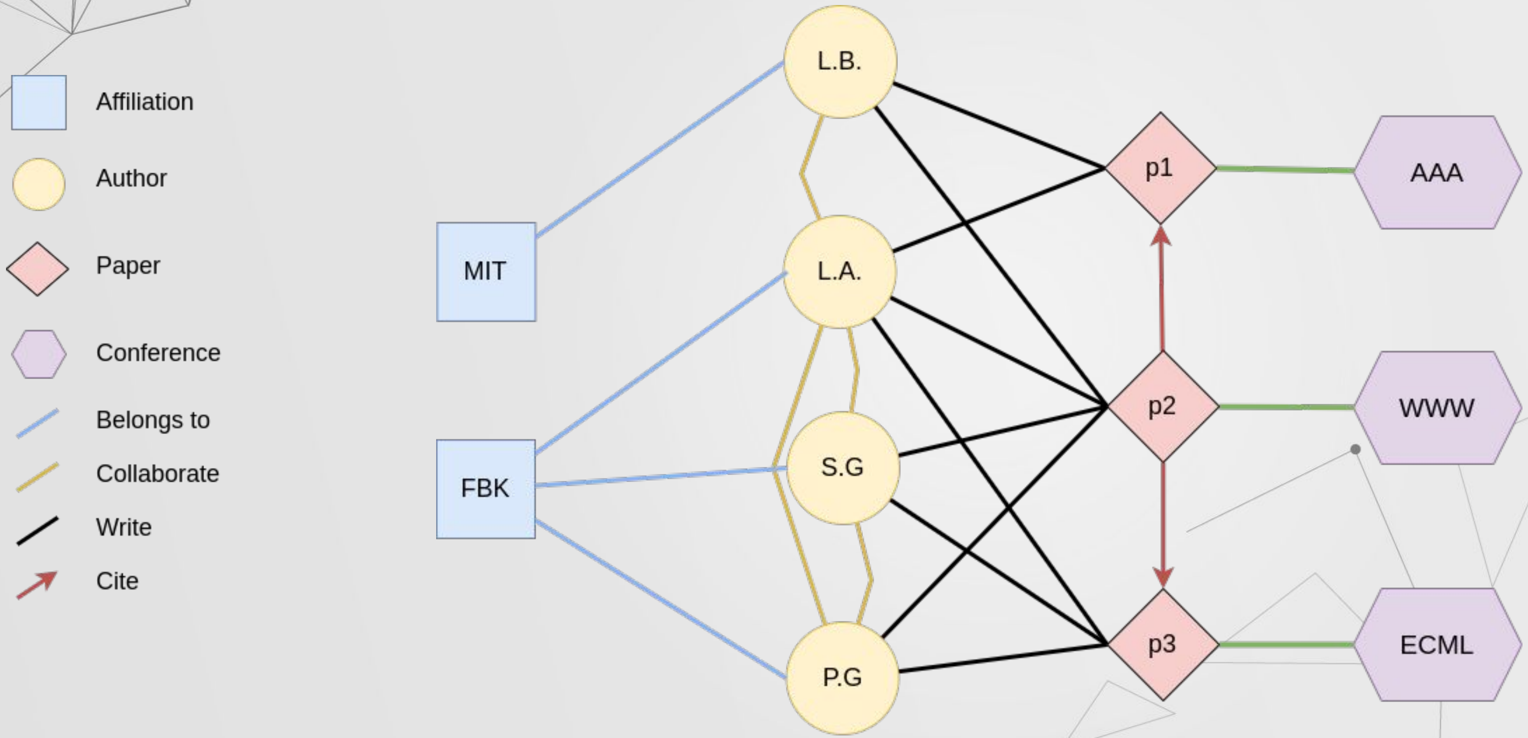
01 Metapath

For instance:



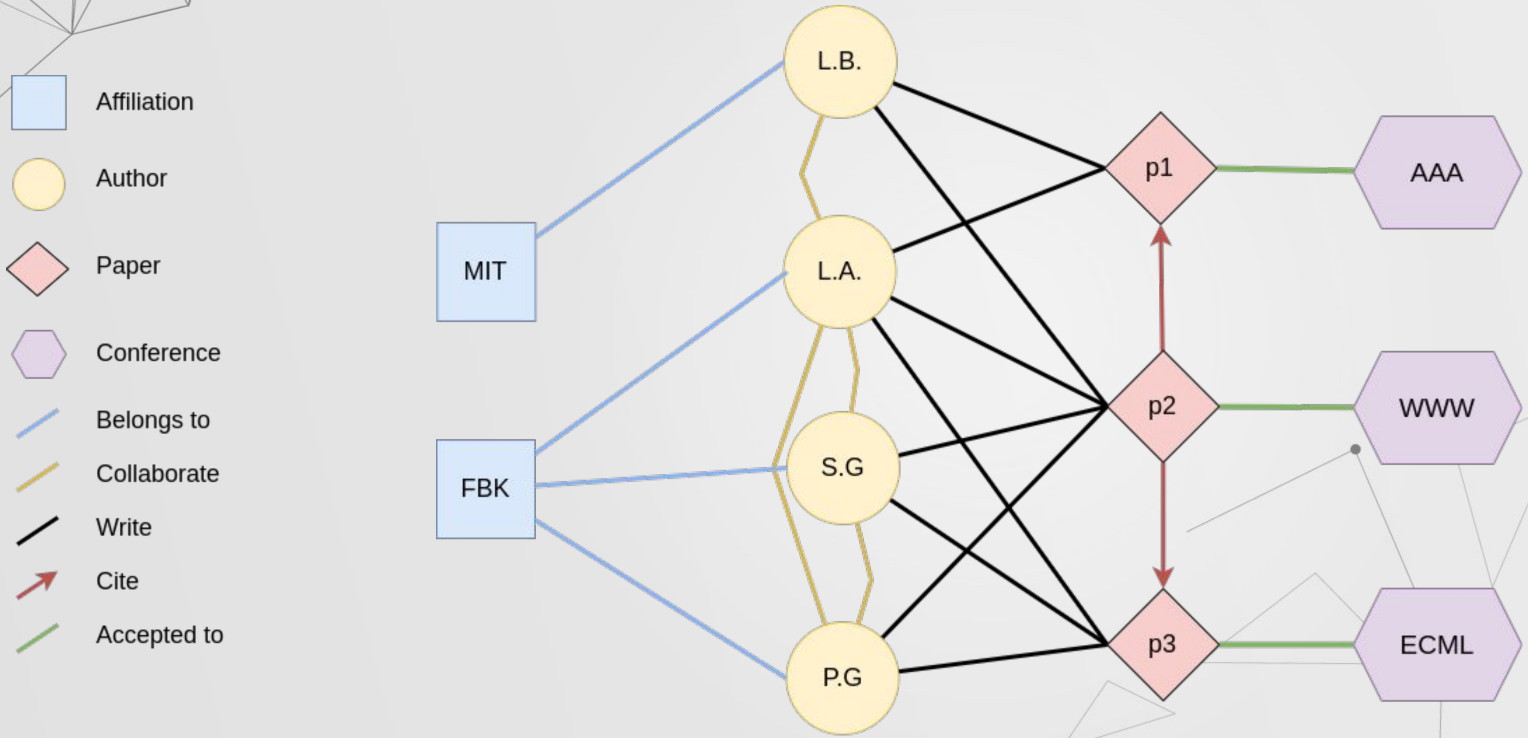
01 Metapath

For instance:



01 Metapath

For instance:



01 Metapath

Def:
A **meta path** in a heterogeneous graphs, is a path following a specific meta path scheme P.

DONG, Yuxiao; CHAWLA, Nitesh V.; SWAMI, Ananthram. metapath2vec: Scalable representation learning for heterogeneous networks. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017. p. 135-144.

01 Metapath

Def:
A **meta path** in a heterogeneous graphs, is a path following a specific meta path scheme P.

Def:
A **meta path scheme** P is defined as a path that is denoted in the form of :

$$V_1 \to^{r_1} V_2 \to^{r_2} V_3 \to^{r_3} V_4 \to^{r_4} \dots V_{l-1} \to^{r_{l-1}} V_l$$

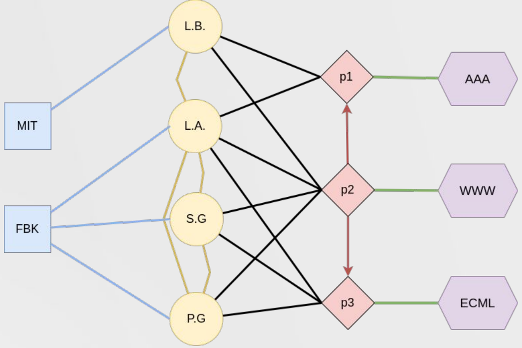
Wherein

$$R = R_1 \cdot R_2 \cdot R_3 \cdot \dots R_{l-1}$$

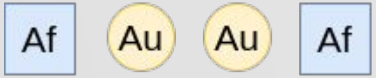
Defines the composite relations between nodes types V_1 and V_l

DONG, Yuxiao; CHAWLA, Nitesh V.; SWAMI, Ananthram. metapath2vec: Scalable representation learning for heterogeneous networks. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017. p. 135-144.

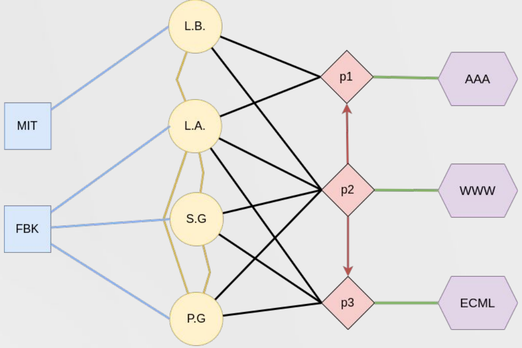
01 Metapath



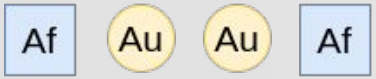
For instance:

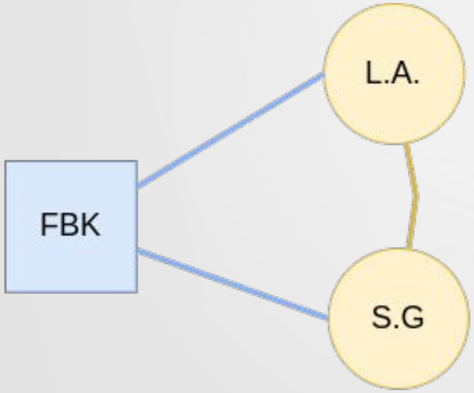


01 Metapath

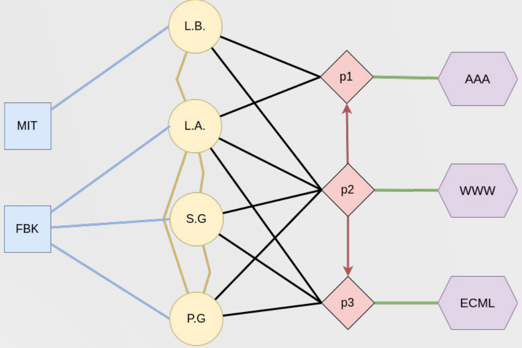


For instance:

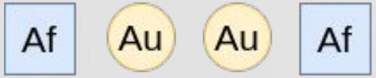


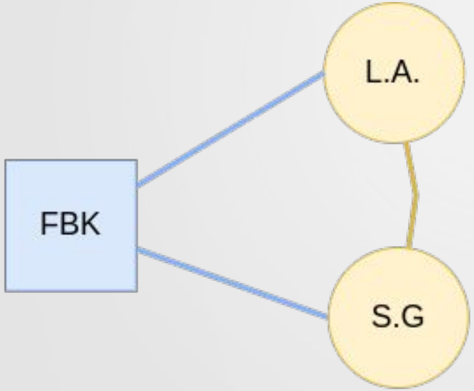


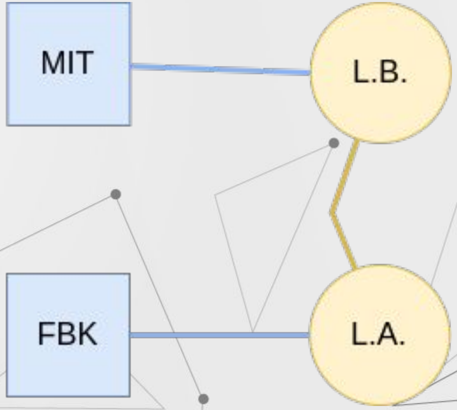
01 Metapath



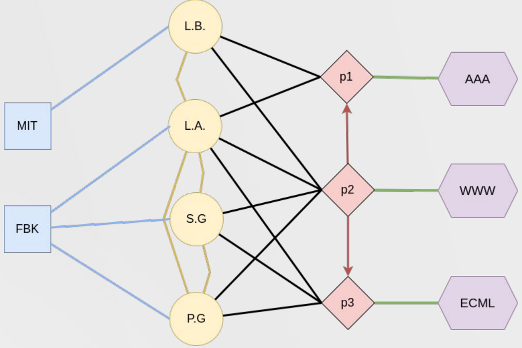
For instance:



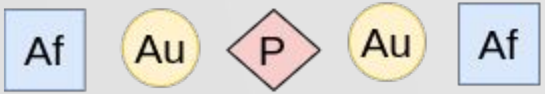




01 Metapath

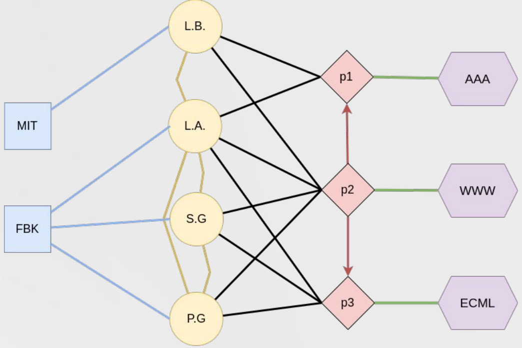


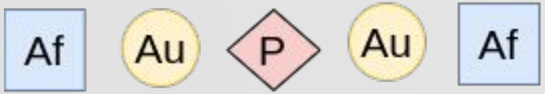
For instance:

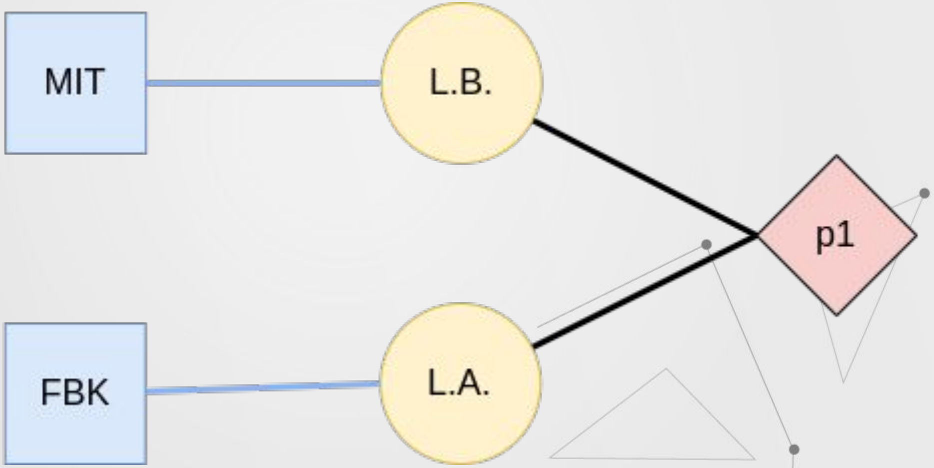


01 Metapath

For instance:







02 Metapath random walks

Def:

Given an heterogeneous network G = (V,E,T) and a meta path scheme P.

The transition probability at step i is defined as follow:

02 Metapath random walks

Def:

Given an heterogeneous network G = (V,E,T) and a meta path scheme P.

The transition probability at step i is defined as follow:

$$p(v^{i+1} | v_t^i, \mathcal{P}) = \begin{cases} \dfrac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) = t{+}1 \\ 0 & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) \neq t{+}1 \\ 0 & (v^{i+1}, v_t^i) \notin E \end{cases}$$

DONG, Yuxiao; CHAWLA, Nitesh V.; SWAMI, Ananthram. metapath2vec: Scalable representation learning for heterogeneous networks. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017. p. 135-144.

02 Metapath random walks

Def:

Given an heterogeneous network G = (V,E,T) and a meta path scheme P.

The transition probability at step i is defined as follow:

Node at time i +1

$$p(v^{i+1} | v_t^i, \mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) = t{+}1 \\ 0 & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) \neq t{+}1 \\ 0 & (v^{i+1}, v_t^i) \notin E \end{cases}$$

02 Metapath random walks

Def:

Given an heterogeneous network G = (V,E,T) and a meta path scheme P.

The transition probability at step i is defined as follow:

Node at time i +1

$$p(v^{i+1}|v_t^i,\mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1},v_t^i) \in E, \phi(v^{i+1}) = t+1 \\ 0 & (v^{i+1},v_t^i) \in E, \phi(v^{i+1}) \neq t+1 \\ 0 & (v^{i+1},v_t^i) \notin E \end{cases}$$

Node at time i
of type t

02 Metapath random walks

Def:

Given an heterogeneous network G = (V,E,T) and a meta path scheme P.

The transition probability at step i is defined as follow:

Node at time i +1

$$p(v^{i+1} | v_t^i, \mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) = t{+}1 \\ 0 & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) \neq t{+}1 \\ 0 & (v^{i+1}, v_t^i) \notin E \end{cases}$$

Node at time i
of type t

Scheme

02 Metapath random walks

Def:

Given an heterogeneous network G = (V,E,T) and a meta path scheme P.

The transition probability at step i is defined as follow:

Function that compute the type of a node

Node at time i +1

$$p(v^{i+1} | v_t^i, \mathcal{P}) = \begin{cases} \frac{1}{|N_{t+1}(v_t^i)|} & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) = t{+}1 \\ 0 & (v^{i+1}, v_t^i) \in E, \phi(v^{i+1}) \neq t{+}1 \\ 0 & (v^{i+1}, v_t^i) \notin E \end{cases}$$

Node at time i of type t

Scheme

03 The model

Metapath2vec

- Given a specific Scheme P

03 The model

Metapath2vec

- Given a specific Scheme P
- Extract random meta path from the input graph

03 The model

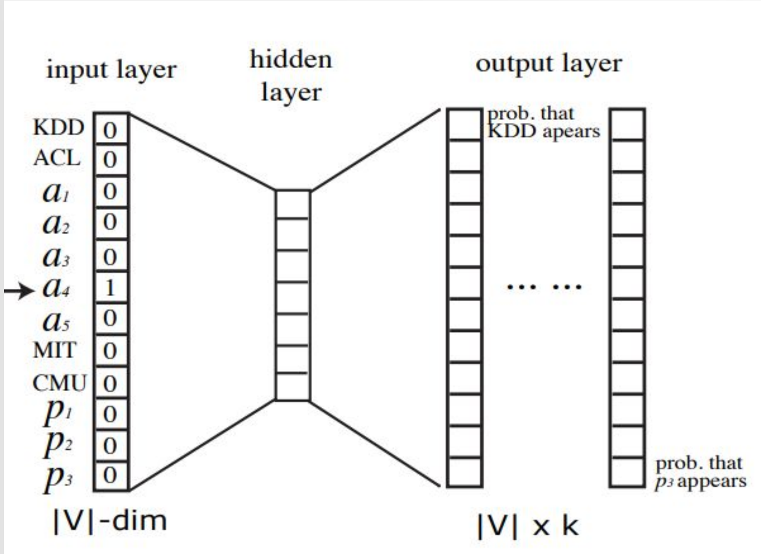
Metapath2vec

- Given a specific Scheme P
- Extract random meta path from the input graph
- Use the skip-gram model

03 The model

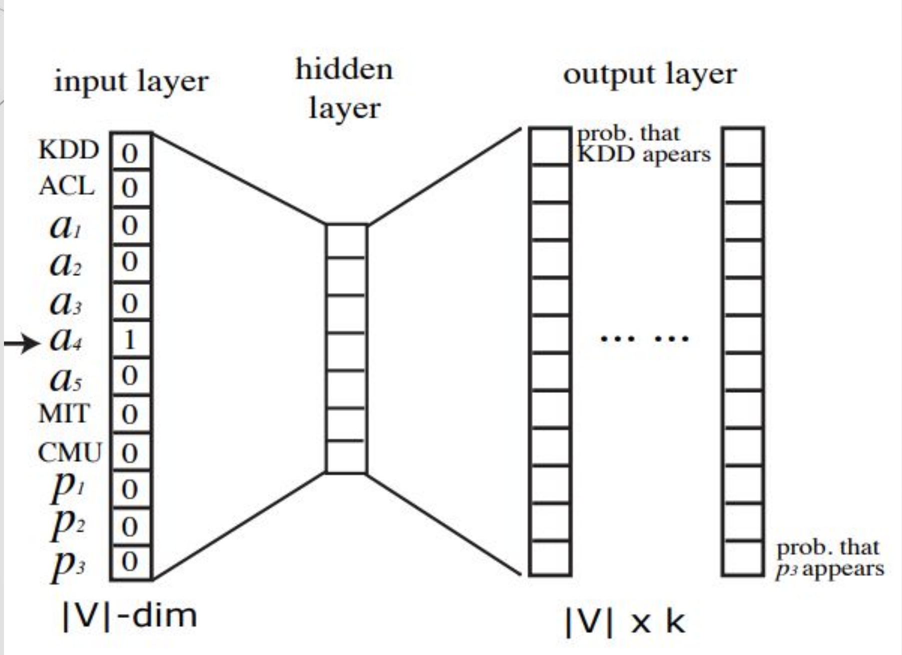
Metapath2vec

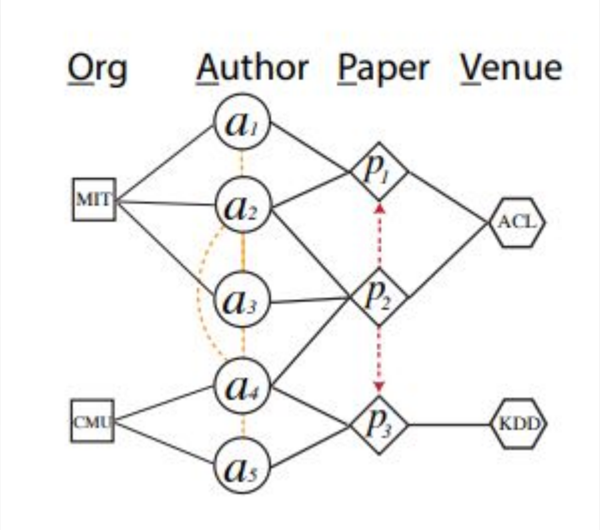
- Given a specific Scheme P
- Extract random meta path from the input graph
- Use the skip-gram model



03 The model

Metapath2vec

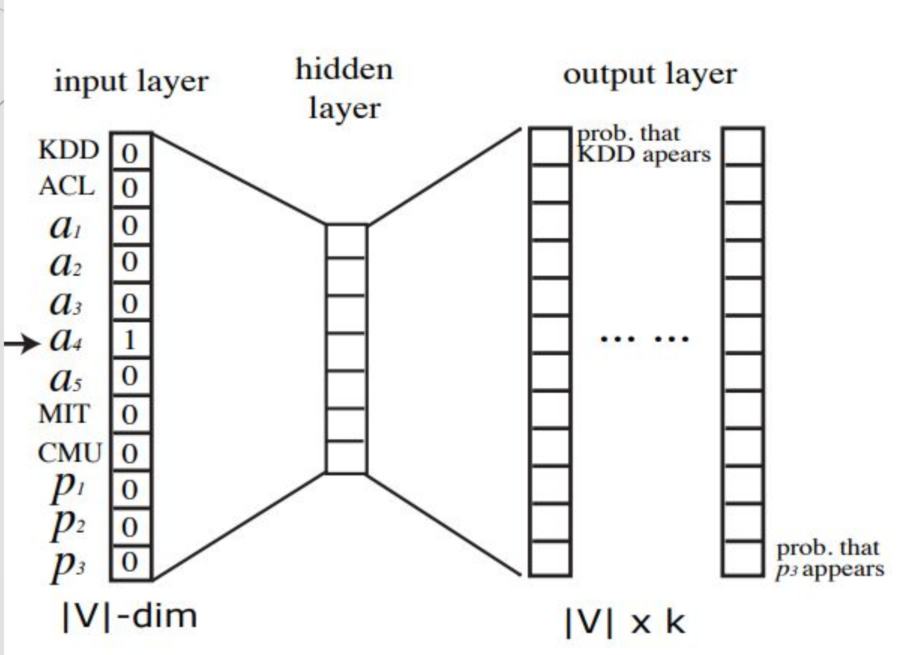


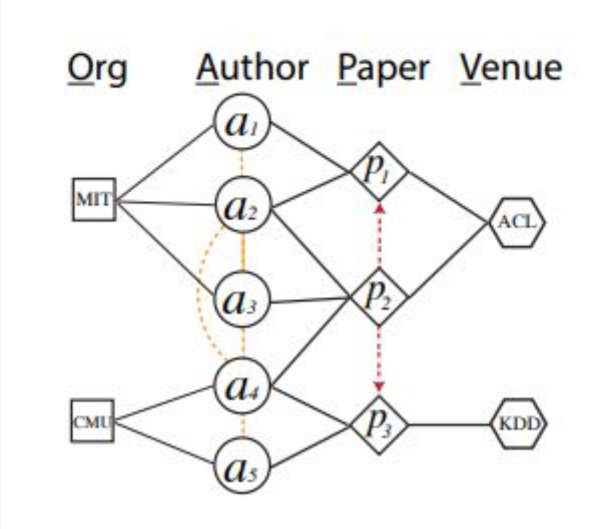


DONG, Yuxiao; CHAWLA, Nitesh V.; SWAMI, Ananthram. metapath2vec: Scalable representation learning for heterogeneous networks. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017. p. 135-144.

03 The model

Metapath2vec

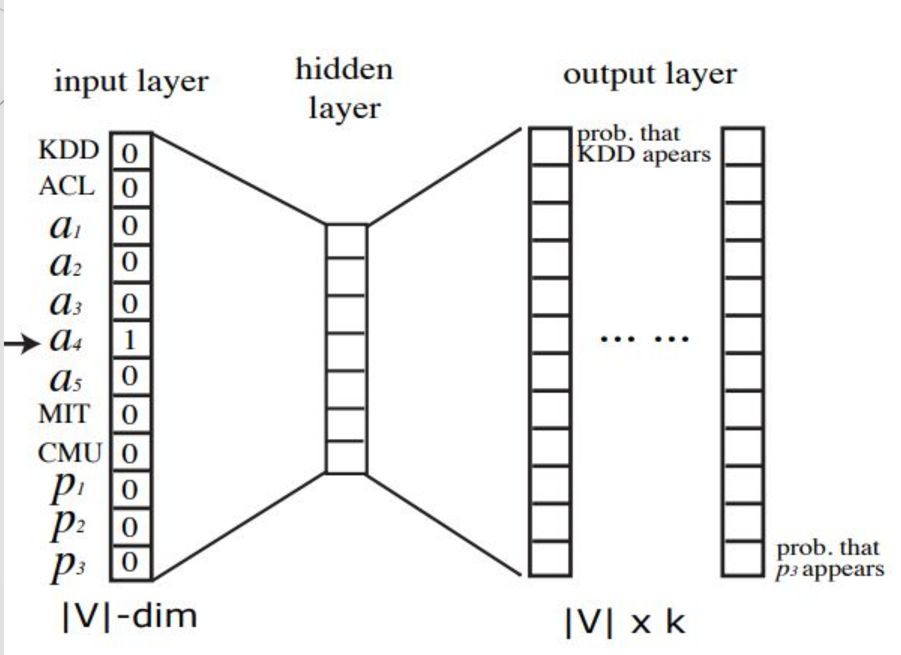


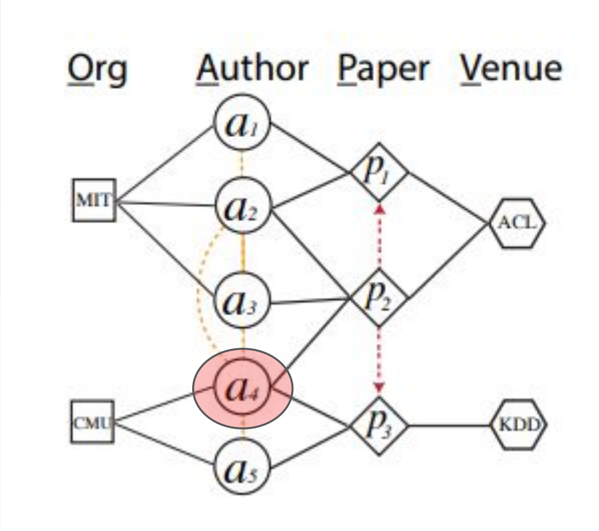


In the case of a4, the output layer is a **12x8,** because...

03 The model

Metapath2vec



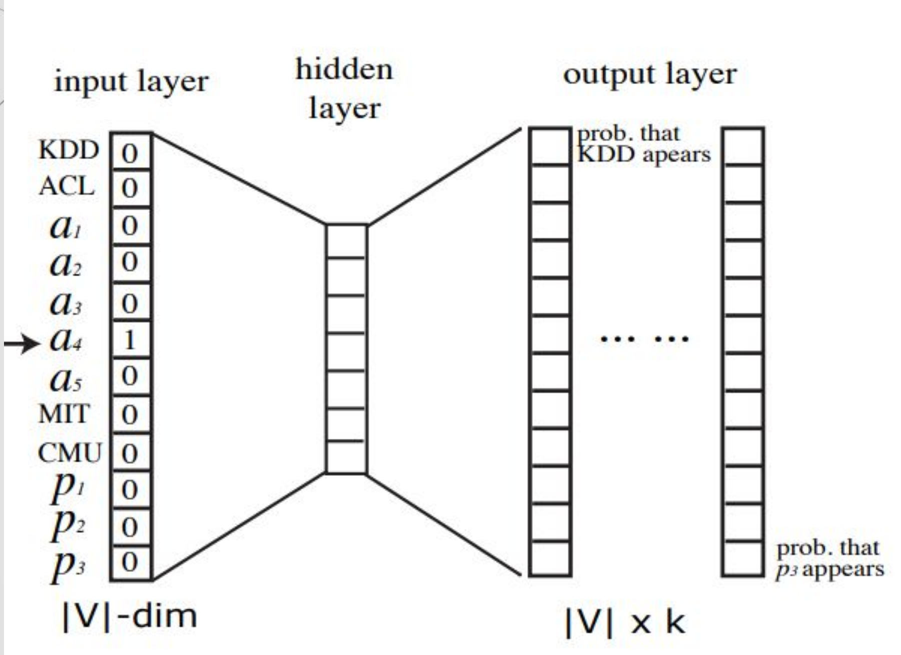


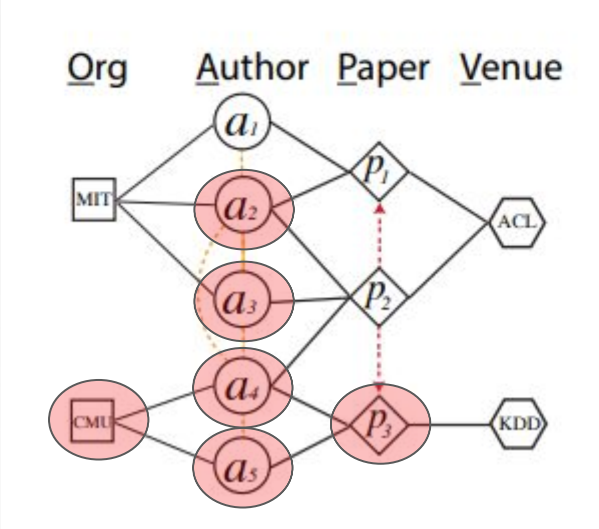
12x0

In the case of a4, the output layer is a **12x8,** because...

03 The model

Metapath2vec



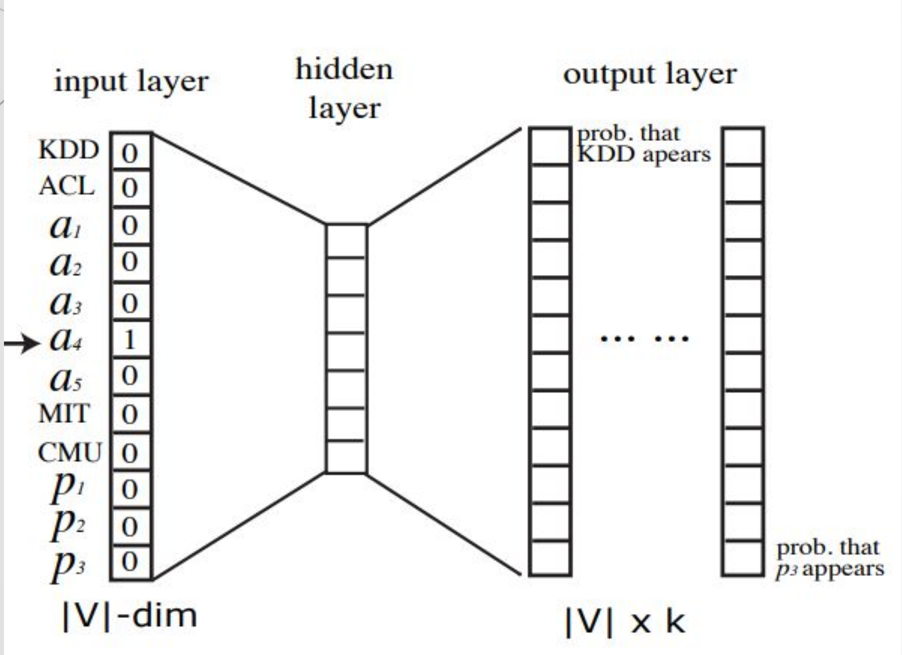


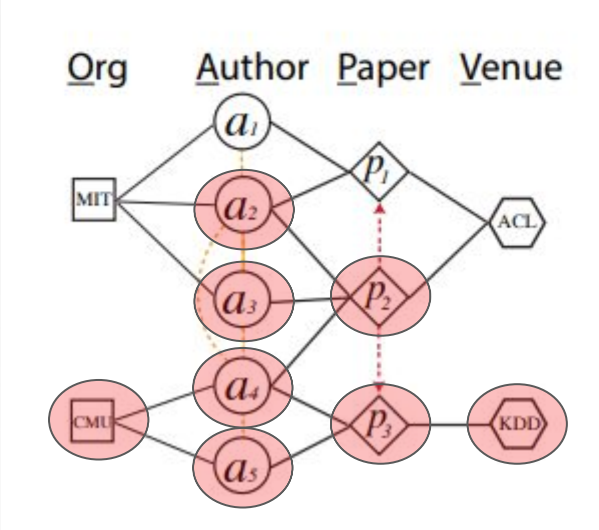
12x5

In the case of a4, the output layer is a **12x8,** because...

03 The model

Metapath2vec



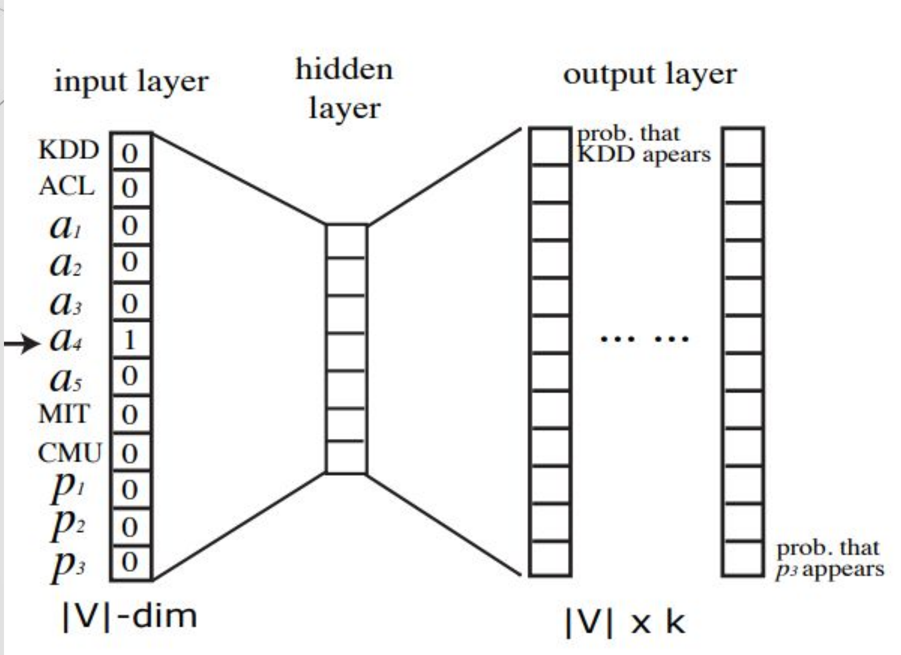


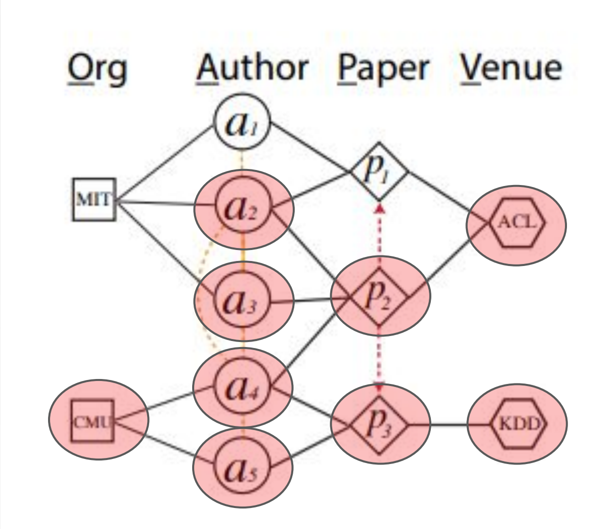
12x7

In the case of a4, the output layer is a **12x8,** because...

03 The model

Metapath2vec





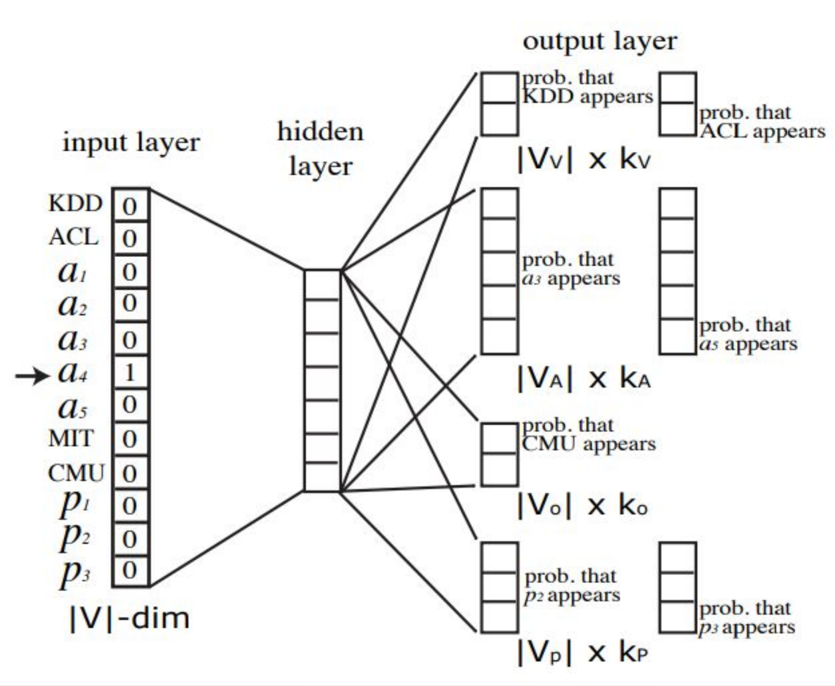
12x8

In the case of a4, the output layer is a **12x8,** because...

03

03 The model

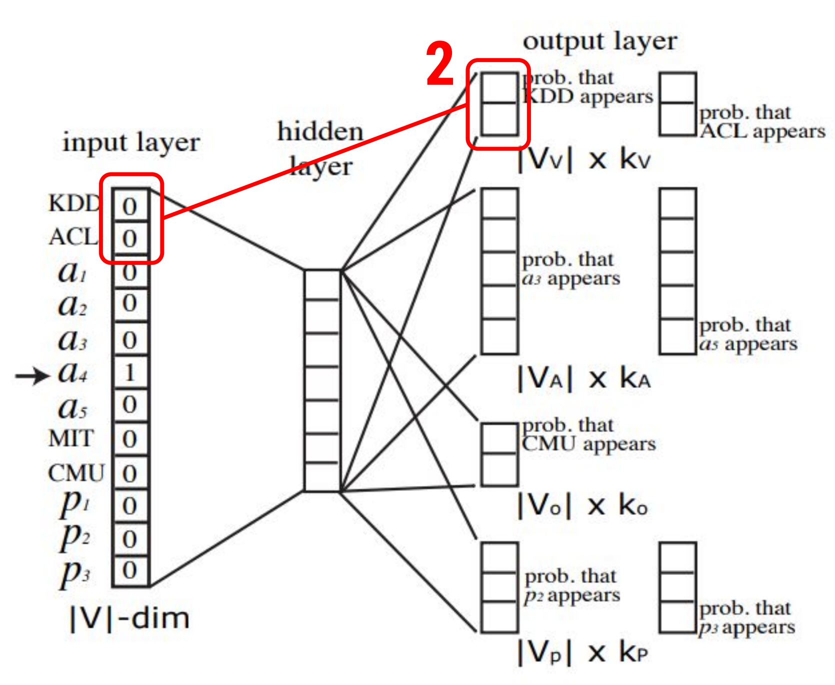
Metapath2vec++



In the case of a4, there are 4 output layers, each of them with different size

03 The model

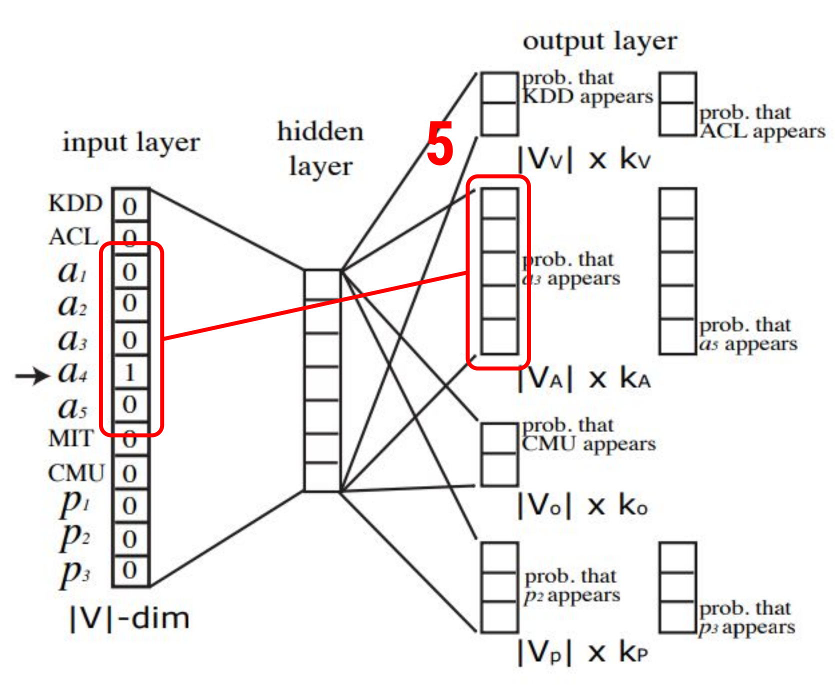
Metapath2vec++



In the case of a4, there are 4 output layers, each of them with different size

03 The model

Metapath2vec++

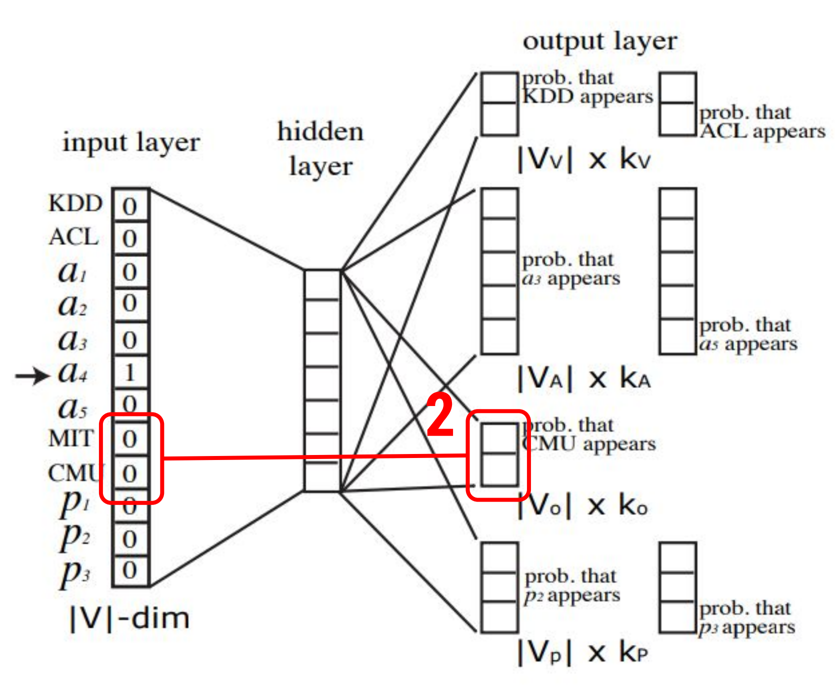


In the case of a4, there are 4 output layers, each of them with different size

DONG, Yuxiao; CHAWLA, Nitesh V.; SWAMI, Ananthram. metapath2vec: Scalable representation learning for heterogeneous networks. In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017. p. 135-144.

03 The model

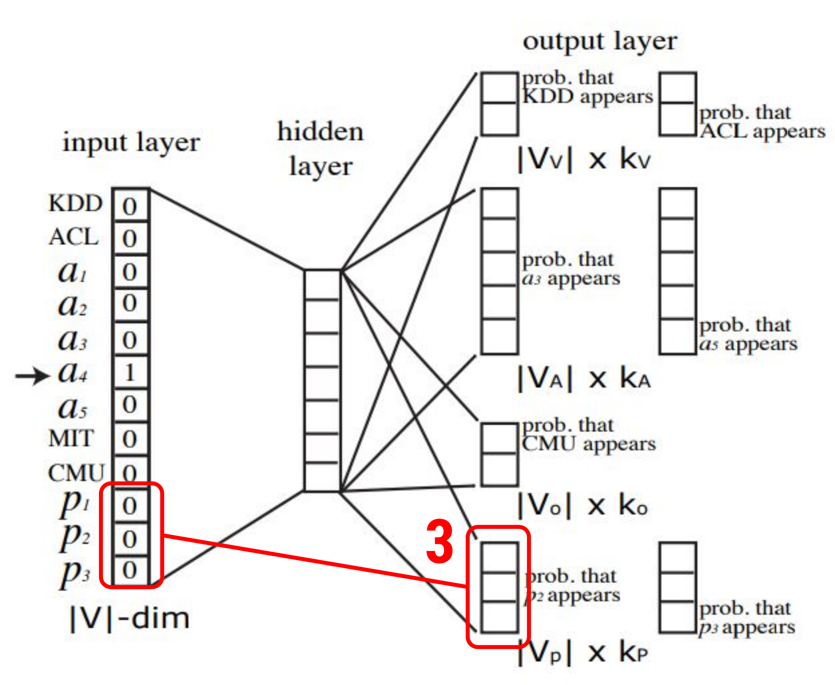
Metapath2vec++



In the case of a4, there are 4 output layers, each of them with different size

03 The model

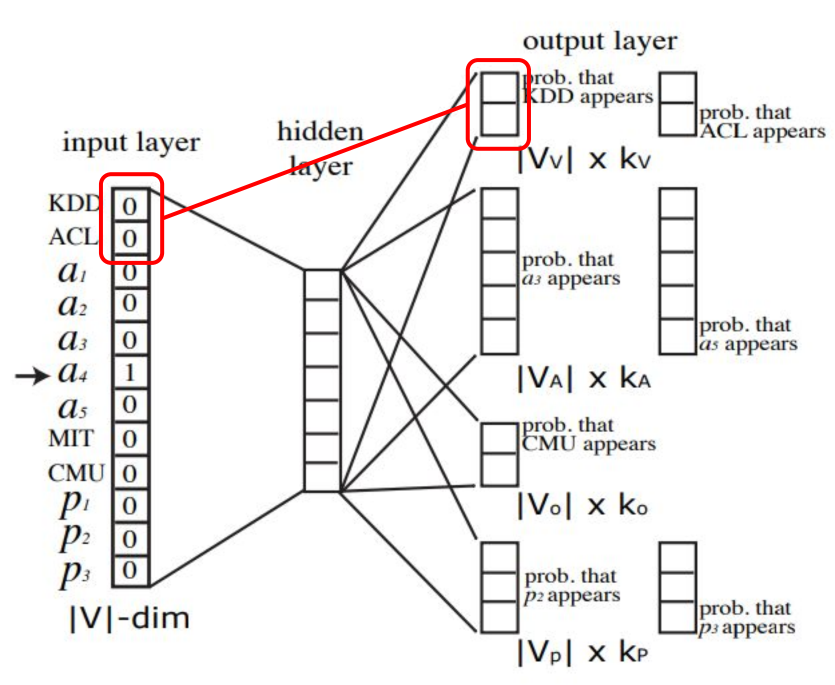
Metapath2vec++



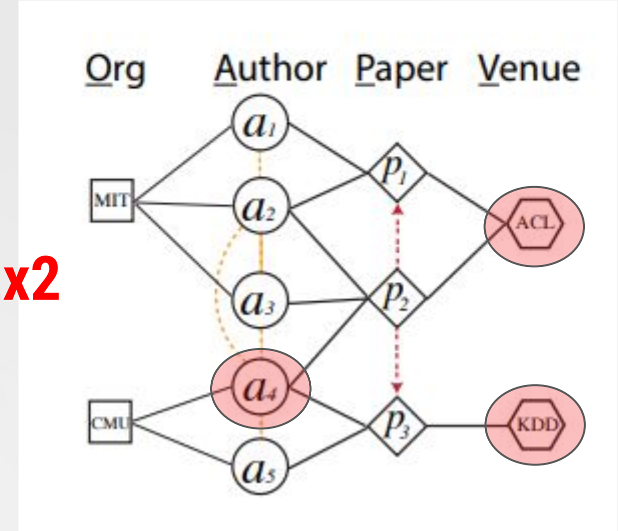
In the case of a4, there are 4 output layers, each of them with different size

03 The model

Metapath2vec++



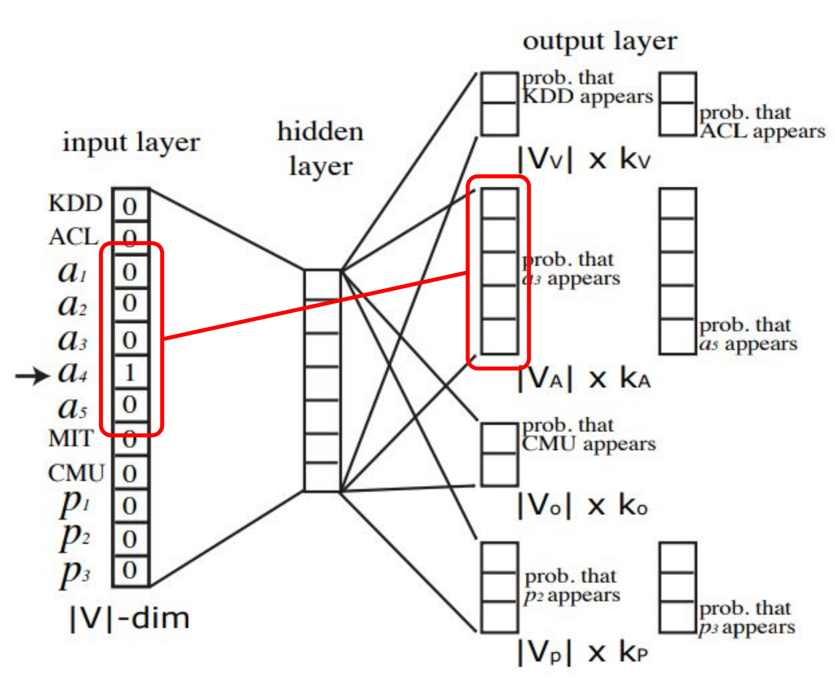
2x2



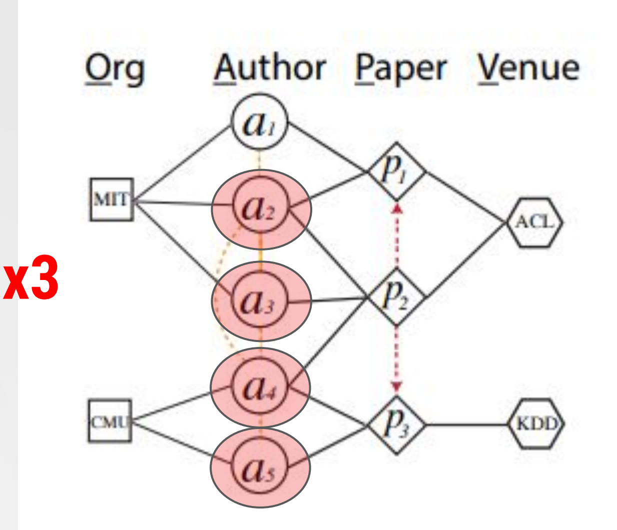
In the case of a4, there are 4 output layers, each of them with different size

03 The model

Metapath2vec++



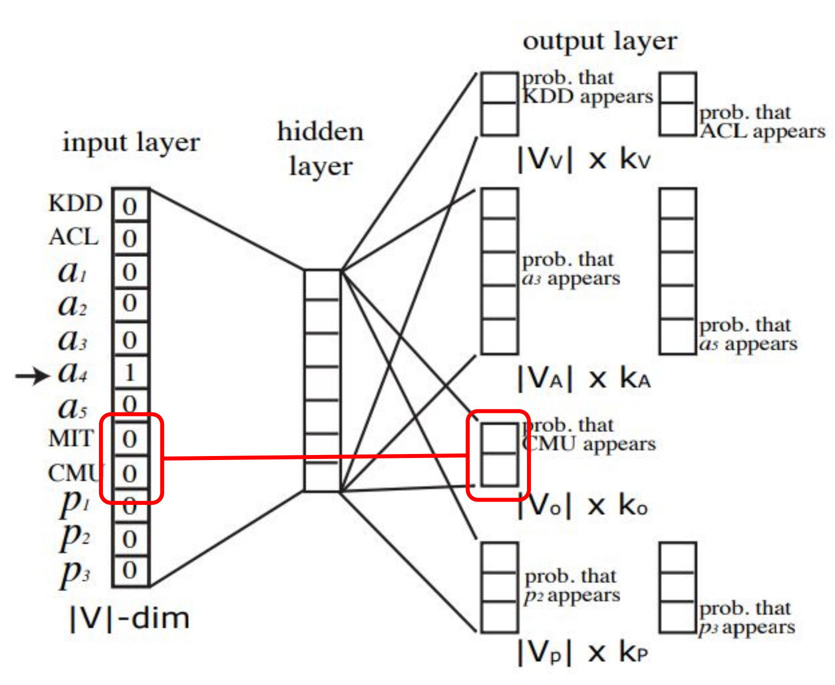
4x3



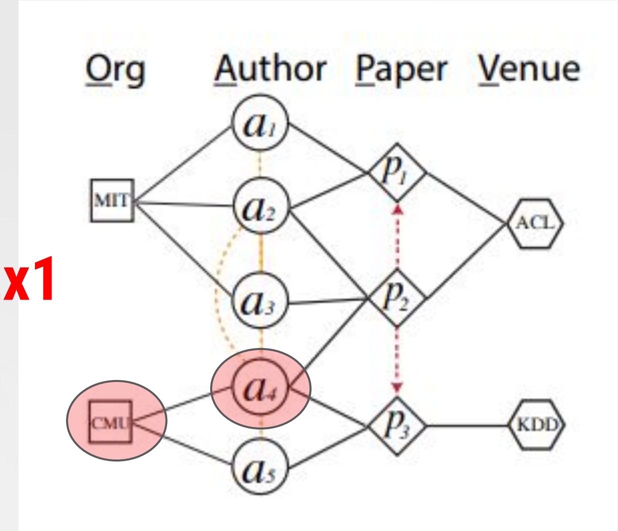
In the case of a4, there are 4 output layers, each of them with different size

03 The model

Metapath2vec++



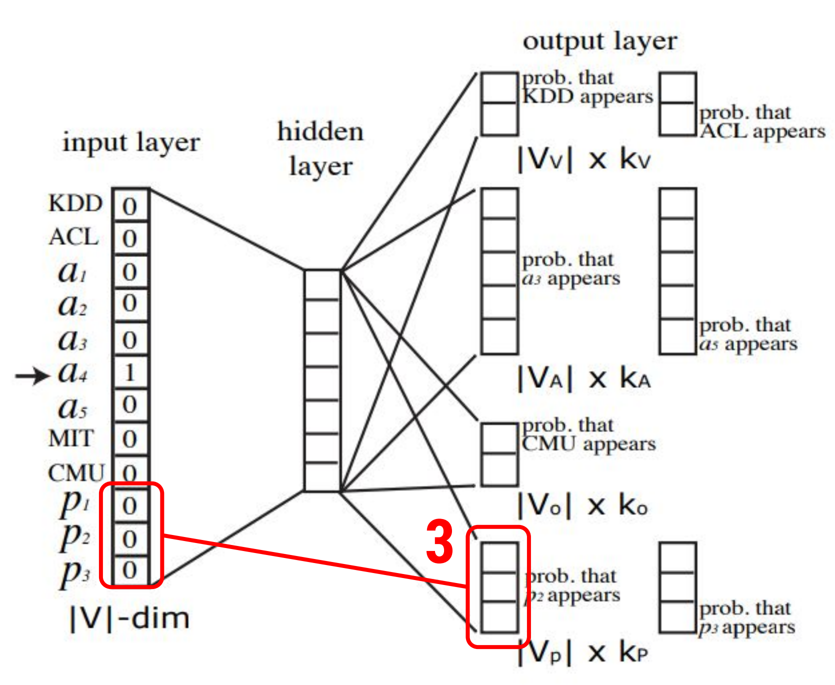
2x1

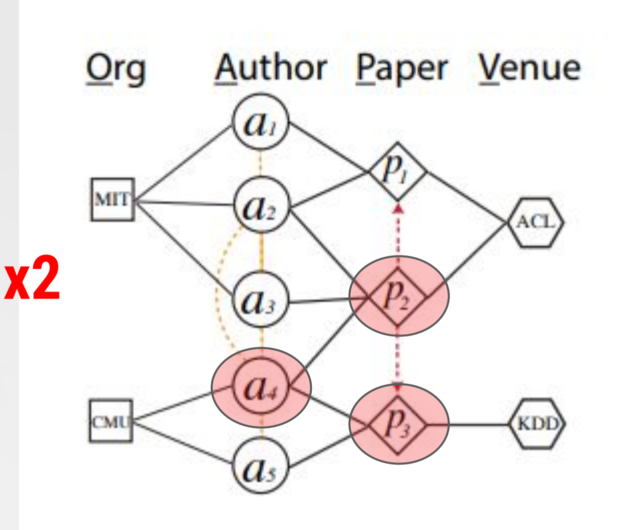


In the case of a4, there are 4 output layers, each of them with different size

03 The model

Metapath2vec++

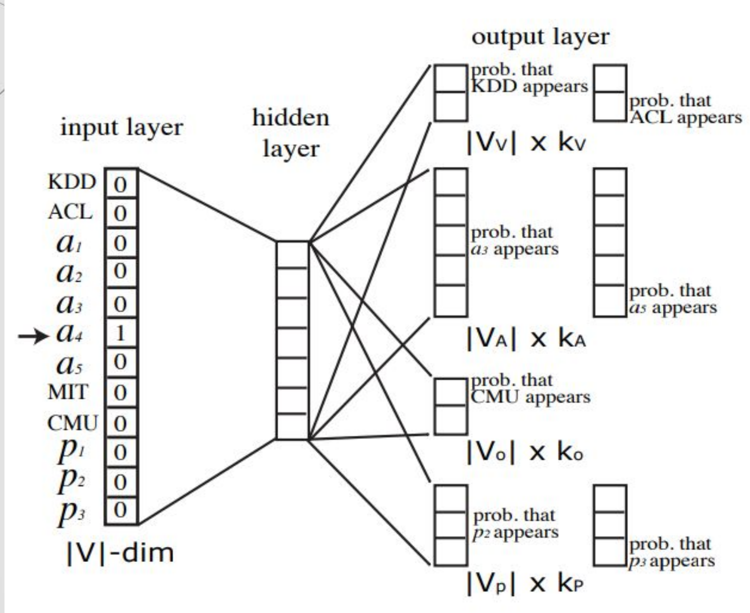


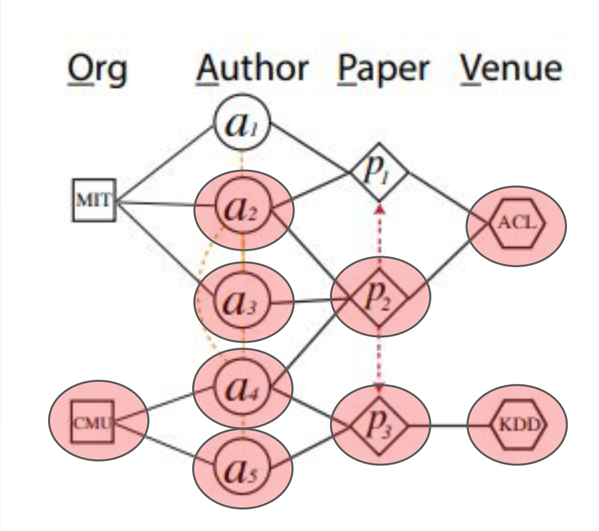


In the case of a4, there are 4 output layers, each of them with different size

03 The model

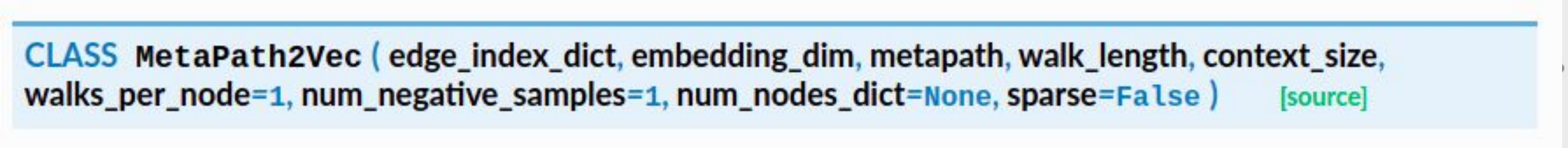
Metapath2vec++





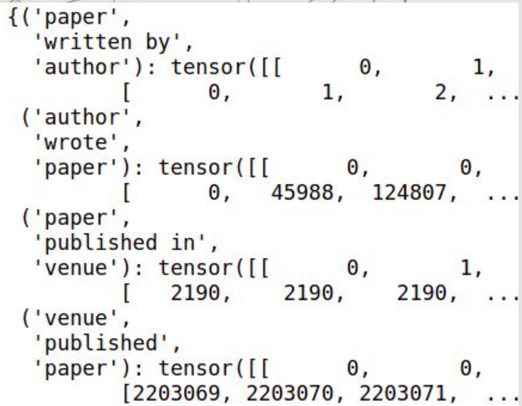
In the case of a4, there are 4 output layers, each of them with different size

04 Training

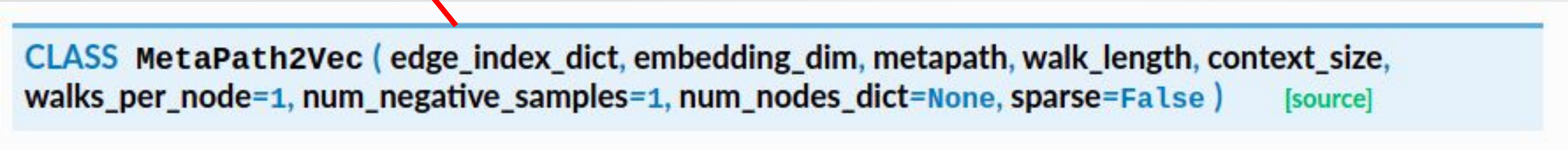


```
CLASS MetaPath2Vec ( edge_index_dict, embedding_dim, metapath, walk_length, context_size,
walks_per_node=1, num_negative_samples=1, num_nodes_dict=None, sparse=False )      [source]
```

04 Training

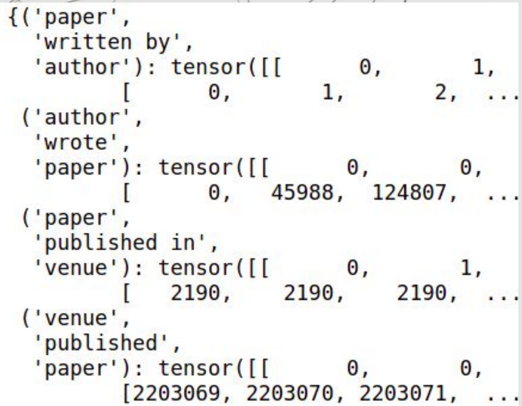


```
{('paper',
  'written by',
  'author'): tensor([[       0,        1,
         [       0,        1,        2,  ...
 ('author',
  'wrote',
  'paper'): tensor([[       0,        0,
         [       0,    45988,   124807,  ...
 ('paper',
  'published in',
  'venue'): tensor([[       0,        1,
         [    2190,     2190,     2190,  ...
 ('venue',
  'published',
  'paper'): tensor([[       0,        0,
         [2203069, 2203070, 2203071,  ...
```



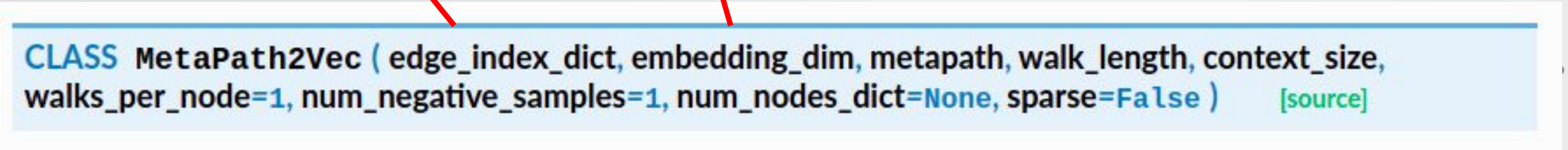
CLASS **MetaPath2Vec** (edge_index_dict, embedding_dim, metapath, walk_length, context_size, walks_per_node=1, num_negative_samples=1, num_nodes_dict=None, sparse=False) [source]

04 Training



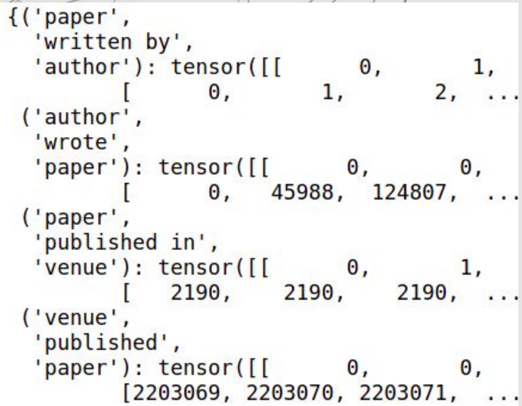
```
{('paper',
  'written by',
  'author'): tensor([[       0,        1,
         [       0,        1,        2,  ...
 ('author',
  'wrote',
  'paper'): tensor([[       0,        0,
         [       0,    45988,   124807,  ...
 ('paper',
  'published in',
  'venue'): tensor([[       0,        1,
         [    2190,     2190,     2190,  ...
 ('venue',
  'published',
  'paper'): tensor([[       0,        0,
         [2203069, 2203070, 2203071,  ...
```

Size of the output embedding



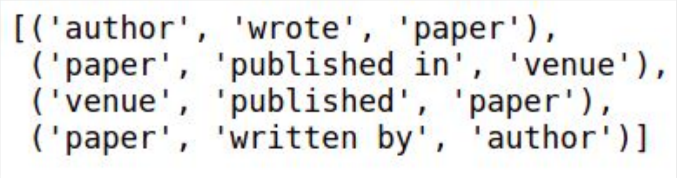
CLASS **MetaPath2Vec** (edge_index_dict, embedding_dim, metapath, walk_length, context_size, walks_per_node=1, num_negative_samples=1, num_nodes_dict=None, sparse=False) [source]

04 Training

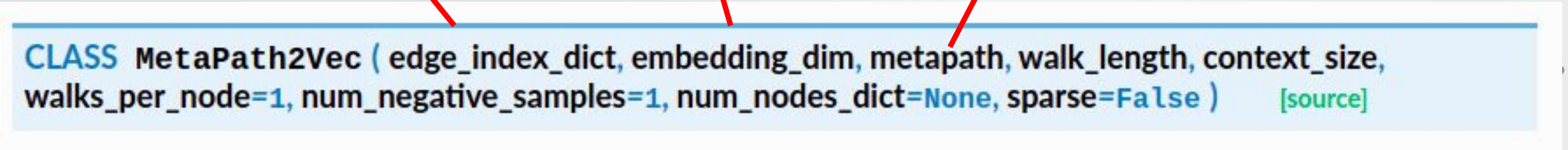


```
{('paper',
  'written by',
  'author'): tensor([[       0,        1,
         [       0,        1,        2,  ...
 ('author',
  'wrote',
  'paper'): tensor([[       0,        0,
         [       0,    45988,   124807,  ...
 ('paper',
  'published in',
  'venue'): tensor([[       0,        1,
         [    2190,     2190,     2190,  ...
 ('venue',
  'published',
  'paper'): tensor([[       0,        0,
         [2203069, 2203070, 2203071,  ...
```

Size of the output embedding

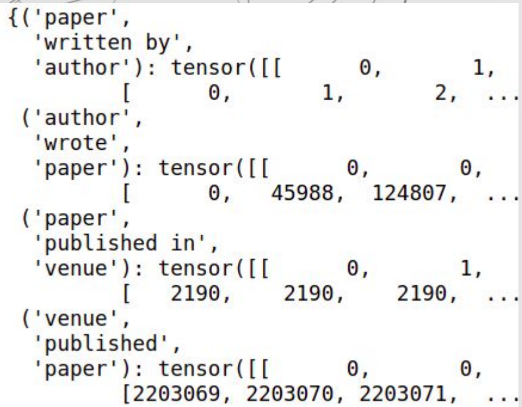


```
[('author', 'wrote', 'paper'),
 ('paper', 'published in', 'venue'),
 ('venue', 'published', 'paper'),
 ('paper', 'written by', 'author')]
```

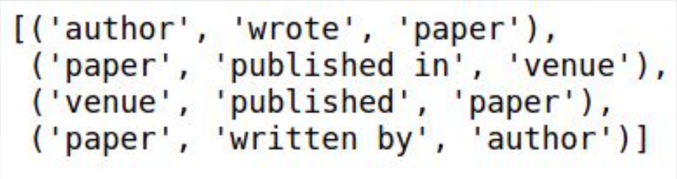


CLASS **MetaPath2Vec** (edge_index_dict, embedding_dim, metapath, walk_length, context_size, walks_per_node=1, num_negative_samples=1, num_nodes_dict=None, sparse=False) [source]

04 Training



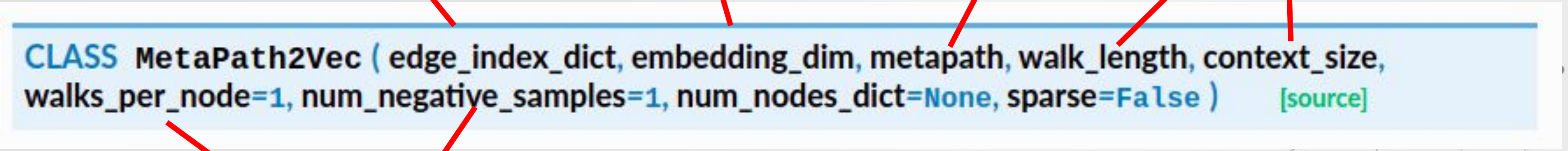
```
{('paper',
  'written by',
  'author'): tensor([[      0,       1,
          [      0,       1,       2,  ...
 ('author',
  'wrote',
  'paper'): tensor([[      0,       0,
          [      0,   45988,  124807,  ...
 ('paper',
  'published in',
  'venue'): tensor([[      0,       1,
          [   2190,    2190,    2190,  ...
 ('venue',
  'published',
  'paper'): tensor([[      0,       0,
          [2203069, 2203070, 2203071,  ...
```



```
[('author', 'wrote', 'paper'),
 ('paper', 'published in', 'venue'),
 ('venue', 'published', 'paper'),
 ('paper', 'written by', 'author')]
```

Size of the output embedding

Equal to Node2Vec



CLASS **MetaPath2Vec** (edge_index_dict, embedding_dim, metapath, walk_length, context_size, walks_per_node=1, num_negative_samples=1, num_nodes_dict=None, sparse=False) [source]

Equal to Node2Vec

04 Training

Jupyter notebook

05 Load a pre trained model

Jupyter notebook